Urban Rail Transit (2023) 9:323–351
https://doi.org/10.1007/s40864-023-00205-1

http://www.urt.cn/

ORIGINAL RESEARCH PAPERS

A Time Series Decomposition and Reinforcement Learning Ensemble Method for Short-Term Passenger Flow Prediction in Urban Rail Transit

Jinxin Wu[1] · Deqiang He[1] · Xianwang Li[1] · Suiqiu He[1] · Qin Li[1] · Chonghui Ren[2]

Received: 29 May 2023 / Revised: 14 September 2023 / Accepted: 15 September 2023 / Published online: 20 November 2023

Abstract Short-term passenger flow prediction (STPFP) helps ease traffic congestion and optimize the allocation of rail transit resources. However, the nonlinear and nonstationary nature of passenger flow time series challenges STPFP. To address this issue, a hybrid model based on time series decomposition and reinforcement learning ensemble strategies is proposed. Firstly, the improved arithmetic optimization algorithm is constructed by adding sine chaotic mapping, a new dynamic boundary strategy, and adaptive T distribution mutations for optimizing variational mode decomposition (VMD) parameters. Then, the original passenger flow data containing nonlinear and nonstationary irregular changes of noise is decomposed into several intrinsic mode functions (IMFs) by using the optimized VMD technology, which reduces the time-varying complexity of passenger flow time series and improves predictability. Meanwhile, the IMFs are divided into different frequency series by fluctuation-based dispersion entropy, and diverse models are utilized to predict different frequency series. Finally, to avoid the cumulative error caused by the direct superposition of each IMF's prediction result, reinforcement learning is adopted to ensemble the multiple models to acquire the multistep passenger flow prediction result. Experiments on four subway station passenger flow datasets proved that the prediction performance of the proposed method was better than all benchmark models. The excellent prediction effect of the proposed model has important guiding significance for evaluating the operation status of urban rail transit systems and improving the level of passenger service.

Keywords Variational mode decomposition · Reinforcement learning · Ensemble strategy · Urban rail transit · Short-term passenger flow prediction

Abbreviations

STPFP	Short-term passenger flow prediction
URT	Urban rail transit
GBDT	Gradient boosting decision tree
BPNN	Back-propagation neural network
LSTM	Long short-term memory
TCN	Temporal convolutional network
EMD	Empirical mode decomposition
EEMD	Ensemble empirical mode decomposition
CEEMDAN	Complete ensemble empirical mode decomposition with adaptive noise
WD	Wavelet decomposition
VMD	Variational mode decomposition
OVMD	Optimized variational mode decomposition
SE	Sample entropy
IMF	Intrinsic mode function
FDE	Fluctuation-based dispersion entropy
PSO	Particle swarm optimization
GWO	Gray wolf optimizer
AOA	Arithmetic optimization algorithm
IAOA	Improved arithmetic optimization algorithm

✉ Deqiang He
hdqlqy@gxu.edu.cn

[1] Guangxi Key Laboratory of Manufacturing System and Advanced Manufacturing Technology, School of Mechanical Engineering of Guangxi University, Nanning 530004, China

[2] Nanning Rail Transit Co., Ltd., Nanning 530029, China

Communicated by Baoming Han.

NAGA-II	Non-dominated sorting genetic algorithm II
QL	Q-learning
OVMD-FDE-QL	Optimized variational mode decomposition-fluctuation-based dispersion entropy-Q-learning
MOA	Math optimizer accelerated
MOP	Math optimizer probability
MSE	Mean squared error
MAE	Mean absolute error
MAPE	Mean absolute percentage error
RMSE	Root mean square error
SDE	Standard deviation of error
CYSS	Chaoyang Square Station
NNERS	Nanning East Railway Station
NNRS	Nanning Railway Station
JHSS	Jinhu Square Station
AFC	Automatic fare collection
DM	Diebold–Mariano

1 Introduction

Urban rail transit (URT) has developed rapidly in recent years and gradually occupied a crucial position in urban public transport. However, with the increasing passenger flow, problems such as passenger congestion and mismatches between supply and demand have attracted increasing attention. Thus, to ensure the stable operation of the URT systems, accurate short-term passenger flow prediction (STPFP) is of great importance in traffic planning [1]. Accurate STPFP results can help URT station managers organize and guide passengers inside the station, ease passenger congestion, and avoid some accidents. For operators, the STPFP results can be used to allocate resources to improve service levels rationally and meet demand for passenger flow. Passengers can also fully grasp the carrying situation of rail transit to improve their travel experience.

The ridership of URT is time-series data continuously recorded and counted by the station card-swiping equipment. Therefore, in earlier studies, time series models were often utilized for STPFP. These models include linear models [2, 3], gray prediction [4], history average [5], and Kalman filter [6]. The premise of these methods is that the time series remains stationary. Still, the traffic situation is changeable in practice, and the passenger flow time-series data are typically complex and nonstationary. As a result, in practical application, these preconditions are challenging to achieve, and the performance of these methods is poor. Thus, some machine learning methods with shallow architecture have been applied to STPFP, such as gradient boosting decision trees (GBDT) [7], support vector machines [8], extreme learning machines [9], and shallow back-propagation neural networks (BPNN) [10]. Although these models can effectively map nonlinear relationships, their performance depends on feature engineering. The production of feature engineering is undoubtedly complex and time-consuming. Therefore, machine learning methods cannot entirely extract the nonlinear characteristics of ridership data. Deep learning based on deep architecture has gained attention in the past few years. It can automatically extract features and has good nonlinear learning abilities. Huang et al. [11] applied deep belief networks to STPFP for different modes of passenger flow. Lv et al. [12] first applied stacked auto-encoders for STPFP. Liu et al. [13] introduced the deep learning theoretical model of recurrent neural networks into transportation. Subsequently, Liu et al. [14] utilized the variant long short-term memory (LSTM) of recurrent neural networks for STPFP. Ma et al. [15] adopted convolutional neural networks to predict large-scale passenger flows. Wu et al. [16] also used graph convolutional neural networks for large-scale STPFP.

However, these single prediction models can only partially cover some influencing factors. The STPFP of the URT stations needs to fully understand the seasonality and trend of passenger flow series and be able to effectively identify impulse noise in the data. URT is a complex system susceptible to heterogeneous interference. The statistical ridership data presents high-level nonlinearity, which makes it challenging for the model to obtain the features of passenger flow [17]. Therefore, some scholars have decomposed the ridership data into several interdependent sub-series using decomposition methods. The sub-series have pronounced apparent trend and fluctuation characteristics, which helps reduce the burden of prediction models [18]. Wei and Chen [19] proposed an empirical mode decomposition (EMD) hybrid model for STPFP, including the transportation time series decomposed by EMD, and formed a hybrid prediction method with a neural network for STPFP. Subsequently, Yang et al. [20] combined EMD and an SAE model for STPFP. Based on EMD, Li and Ma [21] used sample entropy (SE) and EMD to better decrease the sophistication of the time series. Zhang et al. [22] and Liu et al. [23] applied ensemble EMD (EEMD) to decompose ridership data in their research and combined different deep learning models for STPFP. Jia et al. [24] adopted the improved EMD algorithm, CEEMDAN (complete ensemble empirical mode decomposition with adaptive noise), to decompose travel time data. Huang et al. [25] put forward a decomposition and ensemble strategy consisting of CEEMDAN decomposition of original passenger flow data, recurrence quantification analysis reconstruction, and attention mechanism multistep prediction. In addition to EMD and its improved algorithm for decomposing passenger flow time series, wavelet decomposition (WD) is also used for STPFP. Diao et al. [26] proposed a novel Gaussian process model to forecast the detailed components. Liu et al. [27] applied WD to decompose ridership into components of different frequencies, denoise and reconstruct the decomposed data, and finally use a convolutional neural network for STPFP. Yang et al. [28] used WD and deep learning for station passenger flow prediction. However, WD and EMD each have their drawbacks. WD is less adaptive than other modal decomposition methods and must select the parent wavelet to be used in advance. EMD has the disadvantages of mode mixing, endpoint effects, and a lack of a complete theoretical basis. These defects will affect the prediction accuracy of STPFP. Dragomiretskiy and Zosso [29] proposed variational mode decomposition (VMD), which solves the problem of mode mixing and endpoint effect in EMD. Afterward, Li et al. [30] developed a metro passenger flow prediction framework with VMD. Li et al. [31] applied VMD to effectively suppress the influence of randomness and volatility on passenger flow, and the STPFP was carried out accurately. Yang et al. [32] utilized VMD to decompose nonlinear and nonstationary ridership data. They used an extreme learning machine model to correct the prediction errors of each intrinsic mode function (IMF) to complete the prediction. Their experimental results also showed that VMD has a practical application value in STPFP prediction.

The performance of VMD is affected by its parameters. Therefore, the selection of VMD parameters is essential to ensure the result of the hybrid model. Zhang et al. [33] and Fu et al. [34] selected VMD parameters using the center frequency observation method. Their experiments confirmed that this model could enhance the capabilities of hybrid models. But its effect needs to be apparent. With the development of heuristic optimization algorithms, many scholars have applied them to select VMD parameters. For example, genetic algorithms [35], particle swarm optimization (PSO) [36], seagull optimization algorithms [37], and gray wolf optimizers (GWO) [38] have been used. These optimization algorithms generally have the weaknesses of slow convergence speeds, poor stability, and quickly falling into the local optimum, which will affect the VMD parameter optimization. Abualigah et al. [39] put forward a novel arithmetic optimization algorithm (AOA) in 2021, and their experimental results showed that the AOA had surprising advantages in solving challenging optimization problems. Unfortunately, in the optimization process of the AOA, its population diversity will decrease, and it is easiest to fall into a local optimum. An improved AOA (IAOA) is proposed to solve these problems. Sine chaotic mapping, a nonlinear dynamic boundary strategy, and adaptive T distribution mutations are added to enhance the AOA. Afterward, the VMD parameters are optimized by the IAOA.

Many scholars use a fixed machine learning or deep learning model to predict all the decomposed components. Generally, deep learning models are more appropriate for learning high-complexity data, while shallow learning models are more appropriate for stationary series. Although Goh et al. [40] proposed a multimodal approach to predict IMFs, they could have explained more about the complexity criterion of IMFs. Sun and Duan [41] applied sample entropy (SE) to calculate the intricacy of each IMF after decomposition. Similarly, Li et al. [42] adopted SE to analyze the decomposed sub-series. However, SE makes it easy to overlook components that contain helpful information. Azami and Escudero [43] proposed fluctuation-based dispersion entropy (FDE) to process time series. Compared with SE, FDE is more sensitive to time series changes, and the algorithm is more efficient. Thus, the FDE is used in this paper to analyze each IMF's intricacy. According to the FDE results of the sub-series, they are divided into high-frequency and low-frequency series. The models suitable for their respective complexity are used to predict each sub-series.

In addition, most studies usually use direct superposition to reconstruct the prediction results of each IMF. However, they did not consider the cumulative error caused by sub-series superposition. In some other prediction fields, some scholars use ensemble learning methods to integrate multiple different models' prediction results. In applying stacking-based methods [44], Dong et al. [45] applied the stacking-based strategy to integrate the prediction results of five models by taking five prediction models as the base learners. Liu et al. [46] utilized a GBDT to integrate the predictions of the three models. For weight-based ensemble learning methods, some scholars apply heuristic optimization algorithms to obtain the optimal weight of the model. Tian [47] proposed an improved PSO to integrate the predicted values of each prediction mode. Song et al. [48] applied GWO to overcome the limitations of a single prediction model. Moreover, single-objective optimization algorithms can only achieve one criterion. Cheng et al. [49] used a multi-objective salp swarm optimization algorithm to integrate the prediction results of the four basic models. Their optimization objectives include prediction error and model stability. Chen et al. [50] exploited a multi-objective hybrid optimization algorithm with six basic models. The method effectively improved the prediction result. The above studies have shown that stacking-based and weight-based ensemble learning methods perform well. However, the stack-based method belongs to the black box system, and its integration method is incomprehensible. The weight-based ensemble learning method directly assigns weights to each basic model for integration, which is easier to understand than the stacking-based method. Although heuristic algorithms are essential in ensemble learning, making breakthroughs has not been easy. Compared with standard heuristic optimization algorithms, reinforcement learning has a more vital learning ability and has shown strong performance in many fields [51]. Therefore, reinforcement learning is applied to integrate the prediction results of the sub-series prediction model.

In summary, ridership data often exhibit high complexity, making it challenging to improve prediction accuracy. A model called optimized variational mode decomposition-fluctuation-based dispersion entropy-Q-Learning (OVMD-FDE-QL) based on optimized VMD (OVMD), FDE, and ensemble Q-Learning (QL) is proposed for STPFP. Firstly, sine chaotic mapping, a nonlinear dynamic boundary strategy, and adaptive T distribution mutations are added to improve the AOA. The search speed of the AOA is improved by initializing sine chaotic mapping. The nonlinear dynamic boundary strategy is used to improve the local development ability of the AOA in the later stages to balance the global and local search abilities of the algorithm. The adaptive T distribution mutation is applied to prevent the AOA from falling into a local optimal value. Meanwhile, the IAOA is used to find the best parameters of VMD to avoid the shortcomings of VMD selection parameters relying on experience, improving the algorithm's adaptability. Then, the original passenger flow data containing nonlinear and nonstationary irregular noise changes is decomposed into different fluctuation components using OVMD technology, which reduces the time-varying complexity of passenger flow time series and improves predictability. OVMD decomposes the original passenger flow time series to obtain several IMFs, and the FDE is applied to calculate the entropy of each IMF to divide each IMF series into high-frequency and low-frequency series. Higher than a certain threshold are the high-frequency components, and lower than a certain threshold are the stationary low-frequency components. High-frequency IMFs are predicted using LSTMs that are good at processing high-frequency series, and low-frequency IMFs are predicted using back-propagation (BP) neural networks that are good at processing stationary series. Finally, the QL algorithm integrates BP and LSTM to obtain prediction results. The main contributions are as follows:

1. An IAOA algorithm based on sine chaotic mapping, a nonlinear dynamic boundary strategy, and adaptive T distribution mutations is proposed to search the optimal parameters of VMD. The adaptability of the VMD algorithm is enhanced. The experimental results show that the IAOA has better optimization ability and faster convergence speed.
2. The FDE is applied to divide the IMFs into high-frequency and low-frequency series, and the models suitable for their respective complexity are used to predict each sub-series. The experimental results show that FDE can well reflect the complexity of the signal and reasonably divide different frequency components.
3. An ensemble learning method based on reinforcement learning is proposed. The QL method integrates each IMF's results, which avoids the error superposition and blindness of the direct addition of IMFs. The experimental results show that the QL algorithm can effectively integrate the prediction results of each IMF and further improve the prediction results of the model.
4. The ridership of four Nanning Metro stations is applied to certify the performance of OVMD-FDE-QL. Four evaluation indices and four experiments were used for comparison. The experimental results show that OVMD-FDE-QL has strong prediction performance and good versatility.

2 Methodology

2.1 The Passenger Flow Process of OVMD-FDE-QL Model

The framework of the OVMD-FDE-QL STPFP model is shown in Fig. 1. The model includes three modules, and the specific three modules are as follows:

Module 1: Data preprocessing. The IAOA is utilized to optimize the VMD parameters. After obtaining the optimal parameters, the passenger flow data is decomposed into k IMFs by OVMD.

Module 2: Calculate the results of each IMF's FDE and make predictions. Calculate the FDE for each IMF. According to different FDE values, each IMF is split into high-frequency and low-frequency series. The shallow learning BP model is applied to predict the low-frequency component. The deep learning LSTM is utilized to predict high-frequency components.

Module 3: Integrate component prediction results based on the QL. The QL is utilized to integrate predicted values of BP and LSTM. The weighted coefficient w_i is introduced, which is the weighted coefficient of the ith IMF component prediction result. The QL is utilized to optimize the weight coefficients of k IMF components, and a satisfactory result is obtained. Therefore, the output of the ensemble model can be expressed as follows:

$$\hat{O}(t) = w_1\hat{O}_1(t) + w_2\hat{O}_2(t) + \cdots + w_k\hat{O}_k(t), \tag{1}$$

where w_1, w_2, and w_k are the weight coefficients of the first, second, and k IMF components, respectively.$\hat{O}_1(t)$,$\hat{O}_2(t)$, and $\hat{O}_k(t)$ are the prediction results of the first, second, and k IMF components, respectively. In addition, the validation dataset is used to train the QL algorithm. The test dataset is used to evaluate the predictive ability of the OVMD-FDE-QL model.

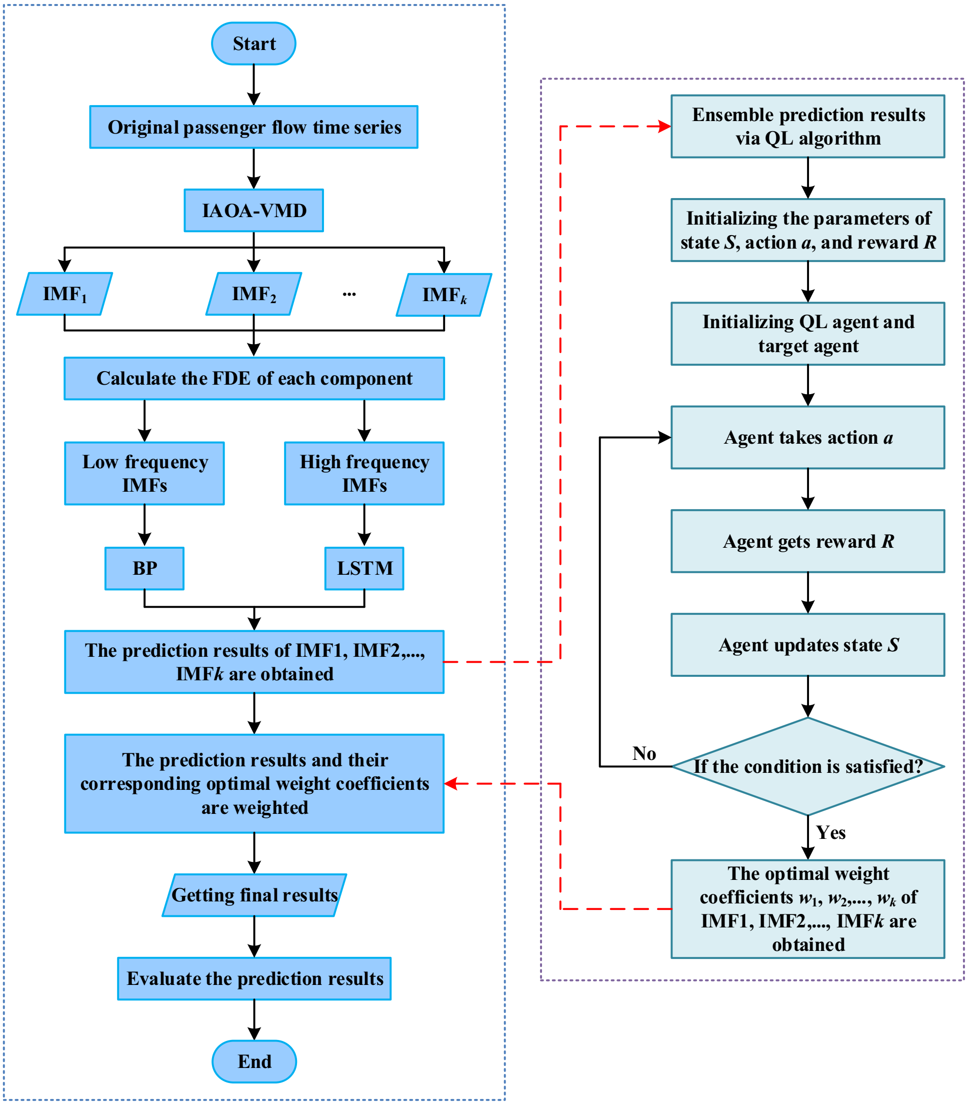

Start
Original passenger flow time series
IAOA-VMD
IMF_1
IMF_2
...
IMF_k
Calculate the FDE of each component
Low frequency IMFs
High frequency IMFs
BP
LSTM
The prediction results of IMF1, IMF2,..., IMF*k* are obtained
The prediction results and their corresponding optimal weight coefficients are weighted
Getting final results
Evaluate the prediction results
End
Ensemble prediction results via QL algorithm
Initializing the parameters of state *S*, action *a*, and reward *R*
Initializing QL agent and target agent
Agent takes action *a*
Agent gets reward *R*
Agent updates state *S*
If the condition is satisfied?
No
Yes
The optimal weight coefficients w_1, w_2,..., w_k of IMF1, IMF2,..., IMF*k* are obtained

Fig. 1 The flowchart of OVMD-FDE-QL

2.2 VMD

VMD is an adaptive, completely non-recursive modal variational and signal processing method. VMD can decompose the signal into k IMFs. Each IMF represents the components of different frequencies and amplitudes, which can better adapt to the characteristics and changes of time series signals. Compared with other traditional decomposition methods, such as WD and EMD, VMD can accurately capture the local characteristics and non-stationarity of signals. In addition, the low coupling of VMD makes it better able to deal with complex time series signals, avoiding the problem of modal aliasing or information loss that may occur in traditional decomposition methods. Therefore, VMD is very suitable for decomposing time-series signals. However, VMD must artificially set the number of modal components k and the penalty factor α [52]. Setting a k value that is too small will make the time series under-decomposed, and a k value that is too large will produce irrelevant modal components. The smaller the α, the greater the noise contained in the decomposed components. The arbitrary setting of these two parameters will affect the effectiveness of VMD decomposition, so selecting the best parameters through an optimization algorithm is necessary.

2.3 Improved AOA Optimized VMD

2.3.1 AOA Algorithm

Abualigah et al. [44] proposed the AOA. They evaluated the AOA's performance, convergence behavior, and computational complexity through different scenarios. Their experimental results showed that the AOA has surprising advantages in solving challenging optimization problems compared to 11 well-known optimization algorithms. Therefore, the AOA is used to select the parameters of

VMD in this paper adaptively. The AOA uses a simple arithmetic operator to calculate the best component in the candidate scheme group, and it generates random numbers through Eq. (2) to initialize the population.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,d} \end{bmatrix}, \tag{2}$$

where N and d denote the number and dimension of solutions, respectively. $x_{i,j}$ is the position of the ith solution in the j-dimensional space. Before the AOA starts optimization, the specific function values of the math optimizer accelerated (MOA) are calculated by Eq. (3) to control the switching between exploration and development of the algorithm.

$$\mathrm{MOA}(t) = \mathrm{Min} + t \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{t_{\max}} \right), \tag{3}$$

where MOA(t) is the specific value of the mathematical optimization coefficient calculated by the current iteration, t is the current number of iterations, and $t_{\max}$ is the maximum number of iterations. The minimum and maximum values of MOA are represented by Min and Max, which are 0.2 and 1, respectively. The random number r_1 with a uniform distribution subject to [0, 1] is selected. The exploration phase is carried out when r_1 is more than MOA. The development phase is performed when r_1 is less than or equal to MOA. The updated expression of the AOA solution in the exploration phase is as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{\mathrm{best},j}^{t} \div (\mathrm{MOP}(t) + \varepsilon) \times ((x_{\mathrm{ub},j} - x_{\mathrm{lb},j}) \times \mu + x_{\mathrm{lb},j}), & r_2 < 0.5 \\ x_{\mathrm{best},j}^{t} \times \mathrm{MOP}(t) \times ((x_{\mathrm{ub},j} - x_{\mathrm{lb},j}) \times \mu + x_{\mathrm{lb},j}), & \text{otherwise} \end{cases}, \tag{4}$$

$$\mathrm{MOP}(t) = 1 - \frac{t^{1/\alpha}}{t_{\max}^{1/\alpha}}, \tag{5}$$

where $x_{i,j}^{t+1}$ represents the position of the ith solution in the jth dimension at the $t+1$ iteration; $x_{\mathrm{best},j}^{t}$ represents the position of the tth-iteration optimal solution in the jth dimension; ε is a small integer, $x_{\mathrm{ub},j}$ and $x_{\mathrm{lb},j}$ represent the upper and lower bounds of the search space on the jth dimension, respectively; u is an adjustable parameter that can reduce the influence of local extremum points on the algorithm optimization, and its value is 0.5; and r_2 is a random number subject to a uniform distribution [0, 1]. When $r_2 < 0.5$, the division operator is used for updating; otherwise, the multiplication operator is used. Math optimizer probability (MOP) is a probability function; α is a sensitivity coefficient, which is used to define the exploration accuracy, and $\alpha = 5$ in the AOA. In the development phase of the AOA, the subtraction search strategy and the addition search strategy are mainly used for development and calculation. The random number r_3 subject to a uniform distribution [0, 1] is selected. If $r_3 < 0.5$, we use the subtraction for optimizing; otherwise, we use the addition for optimizing, which can be expressed as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{\mathrm{best},j}^{t} - MOP(t) \times ((x_{\mathrm{ub},j} - x_{\mathrm{lb},j}) \times \mu + x_{\mathrm{lb},j}), & r_3 < 0.5 \\ x_{\mathrm{best},j}^{t} + \mathrm{MOP}(t) \times ((x_{\mathrm{ub},j} - x_{\mathrm{lb},j}) \times \mu + x_{\mathrm{lb},j}), & \text{otherwise} \end{cases} \tag{6}$$

2.3.2 *Improved AOA*

However, in the optimization process for the AOA, its population diversity will decrease, and it is easy to fall into the local optimum. To solve these problems, an improved AOA (IAOA) is proposed in this paper. Firstly, to improve the diversity of the initial population, sine chaotic mapping [53] is used for population initialization. The formula is as follows:

$$x_{k+1} = \frac{\alpha}{4} \sin(\pi x_k),\ \alpha \in (0, 4]. \tag{7}$$

Then, the coefficient MOA in the AOA is a vital parameter to coordinate the exploration and development phases. However, the linear MOA in Eq. (3) will cause the probability of local development of the algorithm in the later stage of the search to be smaller than that of global exploration. This undoubtedly weakens the local development ability of the AOA in the later stages, which could be more conducive to the optimization speed and accuracy of the algorithm. To improve the AOA's convergence speed and optimization accuracy, a new nonlinear MOA is designed to balance the algorithm's global and local search abilities.

$$\mathrm{MOA}(t) = \mathrm{Min} + (1 - \exp(-(t_{\max} - t)/t_{\max})) \times (\mathrm{Max} - \mathrm{Min}) \tag{8}$$

The new MOA(t) is a nonlinear monotonically decreasing function of t. This enables the algorithm to invest more individuals in global exploration in the early stages of the search. More individuals are invested in local development in the later stages of the algorithmic search. By dynamically adjusting the balance between global search and local search, the precision and stability of the algorithm are improved. Finally, adaptive T distribution [54] mutation ensures the algorithm does not fall into a local optimum in the later iteration stage. The specific mutations are as follows:

$$x_i^t = x_i + x_i \times T(t), \tag{9}$$

where x_i^t is the position of the optimal solution after mutation, x_i is the position of the optimal solution before mutation, and $T(t)$ is the T distribution with the number of iterations of the algorithm as the parameter freedom.

2.3.3 IAOA-VMD

Entropy can be used to evaluate the complexity and sparsity of time series [55]. The smaller the entropy, the better the complexity and sparsity of the time series. This paper uses the envelope entropy as the objective function to optimize VMD. The IAOA is used to obtain the best parameters.

Step 1 Initialization of the IAOA population size, iteration number, dimensions, and the optimal boundary of k and α. In addition, the minimum envelope entropy is used as the fitness function.

Step 2 The population is initialized by sine chaotic mapping, and the fitness value is calculated.

Step 3 Equation (8) is used to calculate the MOA(t), and Eq. (5) is used to calculate the MOP(t). Generate a random number r_1, and find the current optimal individual.

Step 4 If $r_1 < \mathrm{MOA}(t)$, Eq. (4) is used to update the position; otherwise, use Eq. (6) to update the position.

Step 5 Equation (9) is used to carry out adaptive T distribution mutation and retain excellent mutation individuals.

Step 6 Determine whether the algorithm has reached the maximum number of iterations. If the algorithm ends, the best VMD parameters are obtained, or step 2 is returned for re-optimization.

2.4 Sub-series Prediction Models

2.4.1 Long Short-Term Memory Neural Network

LSTM is a temporal recurrent neural network suitable for processing and predicting important events with relatively long intervals and delays in time series [56]. The main difference between LSTM and recurrent neural networks is that LSTM adds a memory cell to judge whether the information is valuable. Three gates are placed in a cell: the forgetting gate (f_t), the input gate (i_t), and the output gate (o_t). The basic structure of LSTM is shown in Fig. 2.

The state of memory cells is controlled by three gates, which can record the history and current system state. The role of the forgetting gate is to control which data in C_{t-1} should be forgotten, which can be expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{10}$$

where h_{t-1} represents the cell's output at the previous moment, x_t represents the input at the current moment, σ is the sigmoid function, and C_{t-1} represents the state of the cell at the previous moment. t represents the time step. W is the weight matrix. b is the bias vector parameter of the neural network. $[h_{t-1}, x_t]$ indicates that the two vectors are connected into a more extended vector.

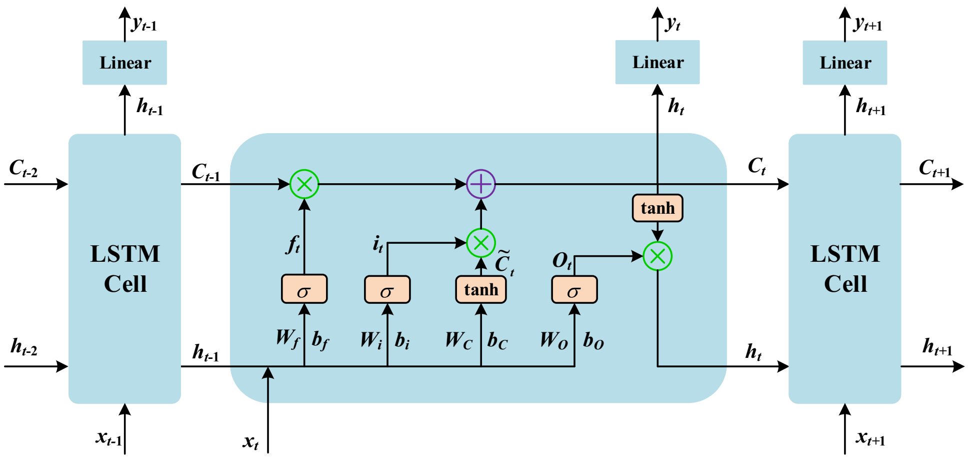
The role of the input gate is to control which data in $\tilde{C}_t$ can be added to the cell state. Specifically, the sigmoid layer determines the information to be updated, while the tanh layer generates a vector added to the state. The formula for the input layer is as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \tag{12}$$

The update of the cell state requires simultaneous action of the forgetting and input gates. The forgetting gate acts on

Fig. 2 The basic structure of LSTM



the previous state C_{t-1}, and the input gate acts on the current input $\tilde{C}_t$. In this way, when a new input a is added to the C_{t-1} after the forgotten gate processing, a new C_t is formed, and an update of the cell state is completed, which can be expressed as follows:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t, \tag{13}$$

where $\otimes$ represents the Hadamard product. The role of the output gate is to determine what value to output. Specifically, the state of the cell is processed by the tanh layer, and the result is multiplied by the sigmoid layer. Finally, the required information is outputted, and the formula of the output gate is as follows:

$$O_t = \sigma(W_O \cdot [h_{t-1}, x_t] + b_O) \tag{14}$$

$$h_t = O_t \otimes \tanh(C_t). \tag{15}$$

In addition, the output at the top is the prediction for the next step's state, achieved through a linear layer, which can be expressed as

$$y_t = W_y h_t + b_y. \tag{16}$$

2.4.2 BP Neural Network

The BPNN is a shallow architecture model composed of an input, hidden, and output layer [57]. Its principle is to adjust the weights and thresholds between the network layers in the form of error reverse propagation to minimize the sum of squared errors. The gradient descent method is usually used as the learning strategy of the BP neural network, and it is very suitable for predicting stationary time series. The modeling process can be expressed as follows:

$$Y = b_1 + \varpi_1(f(b_2 + \varpi_2 \cdot X)), \tag{17}$$

where ω_1 and ω_2 are weights, b_2 and b_1 are biases, and f is the activation function.

2.5 Fluctuation-Based Dispersion Entropy

FDE is a method to measure the complexity and irregularity of time series. FDE is an improved algorithm based on discrete entropy theory. In the calculation process, FDE considers the amplitude difference of adjacent data in the embedded vector and replaces the original data with the difference so that the m-dimensional vector group is compressed into the $m-1$ dimension. After replacement and reduction, the vector model constructed by the original time series can contain local floating changes. Its calculation speed is faster than sample entropy and avoids the problem that permutation entropy ignores signal amplitude. Therefore, FDE is used to evaluate the complexity of the time series in this paper. The calculation steps of FDE are summarized as follows [43]:

(1) Define time series as $x = \{x_j, j = 1, 2, \ldots, N\}$ with length N, and y is mapped to $y = \{y_j, j = 1, 2, \ldots, N\}$. According to the normal distribution function, where $y_j \in (0,\ 1)$ and normal distribution function, y_j is defined as

$$y_j = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x_j} e \frac{-(t-u)^2}{2\sigma^2} dt, \tag{18}$$

where μ and σ represent the expectation and standard deviation of the time series, respectively.

(2) The y is linearly mapped to an integer from 1 to c.

$$z_j^c = \mathrm{round}(c \cdot y_j + \frac{1}{2}), \tag{19}$$

where z_j^c shows the jth member of the classified time series, and rounding involves either increasing or decreasing a number to the next digit. c is the classification number.

(3) Calculate the embedded vector $z_i^{m,c}$:

$$z_i^{m,c} = \{z_i^c, z_{i+d}^c, \ldots, z_{i+(m-1)d}^c\},\ i = 1, 2, \ldots, N-(m-1)d, \tag{20}$$

where m is the embedding dimension, and d is the delay.

(4) The dispersion entropy pattern is defined as $\pi_{v0,\ v1,\ \cdots,\ vm-1}$ $(v = 1, 2, \ldots, c)$. Each $z_i^{m,c}$ is mapped to a dispersion pattern $\pi_{v0,\ v1,\ \cdots,\ vm-1}$ $(v = 1, 2, \ldots, c)$, where $z_i^c = v_0,\ z_{i+d}^c = v_1, \ldots, z_{i+(m-1)d}^c = v_{m-1}$. The number of possible dispersion patterns assigned to each vector $z_i^{m,c}$ is equal to $(2c-1)^{m-1}$, since $z_i^{m,c}$ has m elements, and each can be one of the integers from $-c+1$ to $c+1$.

(5) For each dispersion pattern $\pi_{v0,\ v1,\ \cdots,\ vm-1}$, relative frequency $p(\pi_{v0,\ v1,\ \cdots,\ vm-1})$ is defined as follows:

$$p(\pi_{v0,v1,\ldots,vm-1}) = \frac{N_b(\pi_{v0,v1,\ldots,vm-1})}{N-(m-1)d}, \tag{21}$$

where $N_b(\pi_{v0,\ v1,\ \cdots,\ vm-1})$ represents the number of $z_i^{m,c}$ mapped to $\pi_{v0,\ v1,\ \cdots,\ vm-1}$. Actually, $p(\pi_{v0,\ v1,\ \cdots,\ vm-1})$ shows the ratio of $N_b(\pi_{v0,\ v1,\ \cdots,\ vm-1})$ to $z_i^{m,c}$.

(6) According to the Shannon entropy definition, the FDE of original time series x is defined as

$$\mathrm{FDE}(x, m, c, d) = -\sum_{\pi=1}^{(2c-1)^{m-1}} p(\pi_{v0,v1,\ldots,vm-1})\ln(p(\pi_{v0,v1,\ldots,vm-1})), \tag{22}$$

where x is a time series, m is the embedding dimension, c is the classification number, and d is the delay. This paper sets m, d, and c as 3, 1, and 6, respectively.

As an example, let us have a signal $x = \{2, 3.5, 5.2, 4.1, 2.2, 2.1, 2.5, 4.6, 3.9, 7.4\}$, where we set $d = 1$, $m = 3$, $c = 2$, leading to a $3^2 = 9$ potential dispersion pattern, ({(− 1, − 1), (− 1, 0), (− 1, 1), (0, − 1), (0, 0), (0, 1), (1,− 1), (1, 0), (1, 1)}). Then, x_j($j = 1, 2, \ldots, 10$) are linearly mapped into two classes with integer indices from 1 to 2 ({1, 1, 2, 1, 1, 1, 1, 1, 1, 2}). Afterward, a window with length of 3 moves along the time series, and the differences between adjacent elements are calculated as $x = \{(0, 1), (1, -1), (-1, 0), (0, 0), (0, 0), (0, 0), (0, 0), (0, 1)\}$. Afterward, the number of each dispersion pattern is counted. Finally, using Eq. (22), the FDE value of x is equal to $-\left(\frac{1}{8}\ln\left(\frac{1}{8}\right) + \frac{4}{8}\ln\left(\frac{4}{8}\right) + \frac{2}{8}\ln\left(\frac{2}{8}\right) + \frac{1}{8}\ln\left(\frac{1}{8}\right)\right) = 1.2130.$

2.6 Ensemble Reinforcement Learning Method

Reinforcement learning is different from supervised and unsupervised learning. It learns through agents in a trial-and-error way. Compared with the stacking-based and optimization-based ensemble methods, reinforcement learning has a more vital ensemble learning ability. Learning information is obtained by receiving environment rewards for actions, and model parameters are updated [58]. Q-learning (QL) is a value-based learning algorithm in reinforcement learning [59]. QL creates a Q table for the relationship between state and action, and the value of the Q table is the reward for each action taken in each state. The algorithm is used to combine the prediction results of the k-modal components. The QL algorithm obtains the optimal ensemble weights of each modal component to minimize the reconstruction error of the modal component. The specific steps are as follows:

Step 1 Initialize QL's state S and various parameters, including the reward discount factor γ, learning rate α, greedy parameter ε, and Q table Q(S, a). The state matrix $S = [w_1, w_2, \ldots, w_k]$, and action matrix $a = [\Delta w_1, \Delta w_2, \ldots, \Delta w_k]$. w represents the weight of each sub-series, and Δw is the weight change.

Step 2 Choose a according to the ε-greedy strategy. Its mathematical expression is as follows:

$$a_n = \begin{cases} randomly\ action, probability\ \varepsilon \\ based\ on\ Q_{\max}, probability\ 1 - \varepsilon \end{cases} \tag{23}$$

Step 3 Establish a loss function L and a reward mechanism R. The role of the reward is to allow the agent to minimize the prediction error, and the mean squared error (MSE) is used as the loss function:

$$L = \frac{1}{N}\sum_{t=1}^{N}[\hat{O}(t) - O(t)]^2, \tag{24}$$

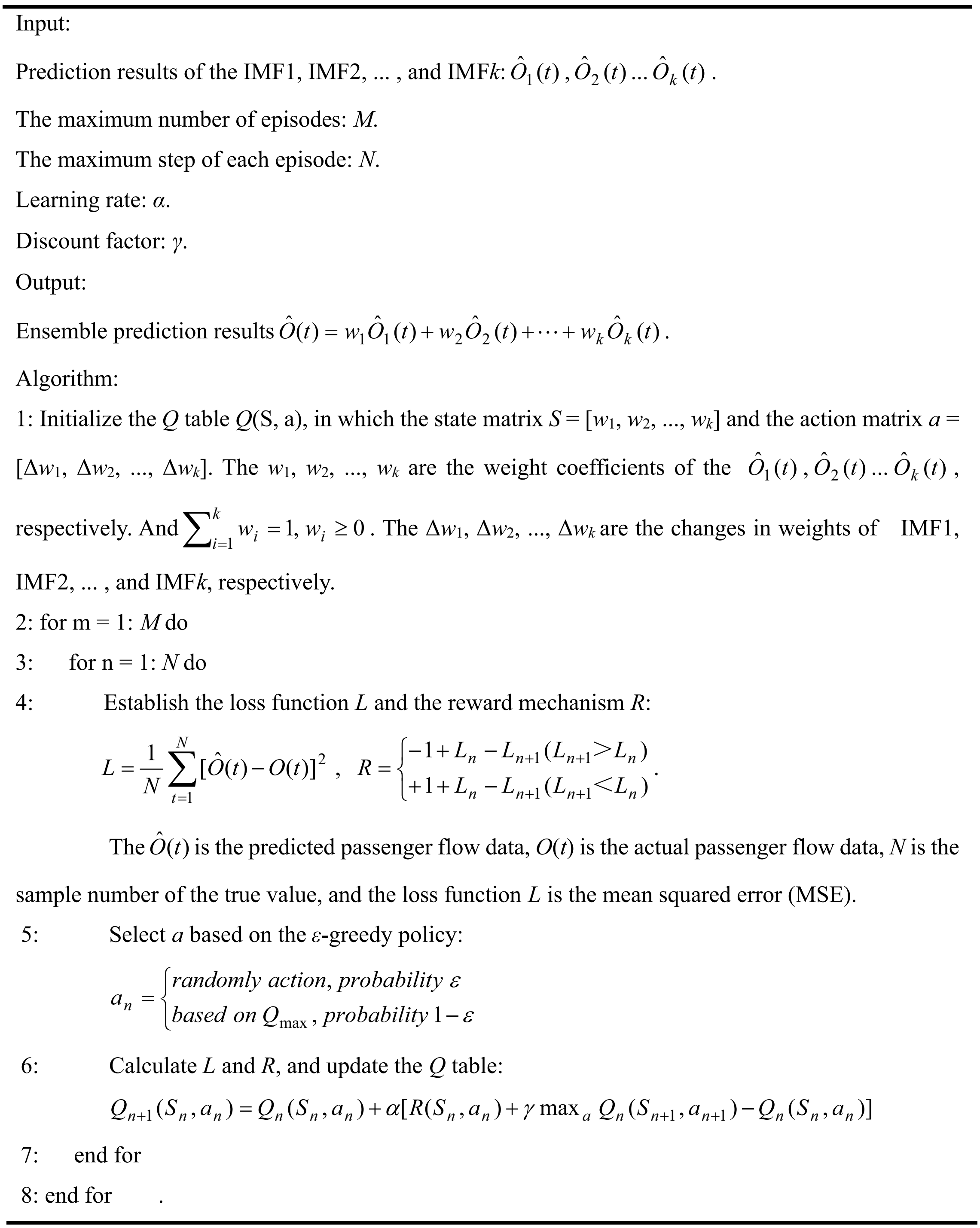
where $\hat{O}(t)$ is the predicted passenger flow data, $O(t)$ is the actual passenger flow data, and N is the sample number of the actual value. The agent's reward is determined by the MSE obtained by the model taking action. The rewards can be expressed as

$$R = \begin{cases} -1 + L_n - L_{n+1}(L_{n+1} > L_n) \\ +1 + L_n - L_{n+1}(L_{n+1} < L_n) \end{cases} \tag{25}$$

Step 4 Calculate L and R and update the Q table. The updated mathematical formula is (26):

$$\begin{aligned} Q_{n+1}(S_n, a_n) =& Q_n(S_n, a_n) + \alpha[R(S_n, a_n) \\ &+ \gamma \max_a Q_n(S_{n+1}, a_{n+1}) - Q_n(S_n, a_n)]. \end{aligned} \tag{26}$$

Step 5 Repeat steps 2–4 until the iteration termination condition is met. Finally, the optimal weight of each sub-series is obtained, and the final output of the integrated model is obtained according to Eq. (1). The right half of Fig. 1 shows the reinforcement learning ensemble strategy process. The specific steps of the ensemble prediction method based on the QL algorithm are shown in Algorithm 1.



```
Input:
Prediction results of the IMF1, IMF2, ... , and IMFk: $\hat{O}_1(t)$, $\hat{O}_2(t)$ ... $\hat{O}_k(t)$ .
The maximum number of episodes: M.
The maximum step of each episode: N.
Learning rate: α.
Discount factor: γ.
Output:
Ensemble prediction results $\hat{O}(t) = w_1\hat{O}_1(t) + w_2\hat{O}_2(t) + \cdots + w_k\hat{O}_k(t)$ .
Algorithm:
1: Initialize the Q table Q(S, a), in which the state matrix $S = [w_1, w_2, ..., w_k]$ and the action matrix $a = [\Delta w_1, \Delta w_2, ..., \Delta w_k]$. The $w_1, w_2, ..., w_k$ are the weight coefficients of the $\hat{O}_1(t)$, $\hat{O}_2(t)$ ... $\hat{O}_k(t)$, respectively. And $\sum_{i=1}^{k} w_i = 1, w_i \geq 0$ . The $\Delta w_1, \Delta w_2, ..., \Delta w_k$ are the changes in weights of IMF1, IMF2, ... , and IMFk, respectively.
2: for m = 1: M do
3:     for n = 1: N do
4:         Establish the loss function L and the reward mechanism R:
           $L = \frac{1}{N}\sum_{t=1}^{N}[\hat{O}(t) - O(t)]^2$ ,  $R = \begin{cases} -1 + L_n - L_{n+1} (L_{n+1} > L_n) \\ +1 + L_n - L_{n+1} (L_{n+1} < L_n) \end{cases}$.
           The $\hat{O}(t)$ is the predicted passenger flow data, $O(t)$ is the actual passenger flow data, N is the sample number of the true value, and the loss function L is the mean squared error (MSE).
5:         Select a based on the ε-greedy policy:
           $a_n = \begin{cases} randomly\ action, probability\ \varepsilon \\ based\ on\ Q_{\max}, probability\ 1-\varepsilon \end{cases}$
6:         Calculate L and R, and update the Q table:
           $Q_{n+1}(S_n, a_n) = Q_n(S_n, a_n) + \alpha[R(S_n, a_n) + \gamma \max_a Q_n(S_{n+1}, a_{n+1}) - Q_n(S_n, a_n)]$
7:     end for
8: end for       .
```

Algorithm 1. Ensemble strategy method based on QL algorithm

3 Experiments and Discussion

Four different types of URT station passenger flow data are tested and analyzed to certify the predictive ability and versatility of the proposed model. In addition, the ablation experiments are also carried out to certify OVMD-FDE-QL.

Table 1 The features of the original passenger flow data

Datasets	Numbers	Mean	Standard deviation	Max	Min
		Passengers/15 min	Passengers/15 min	Passengers/15 min	Passengers/15 min
CYSS	2415	517	331	1892	1
NNERS	2415	303	184	1103	1
NNRS	2415	280	138	752	2
JHSS	2415	140	136	859	1

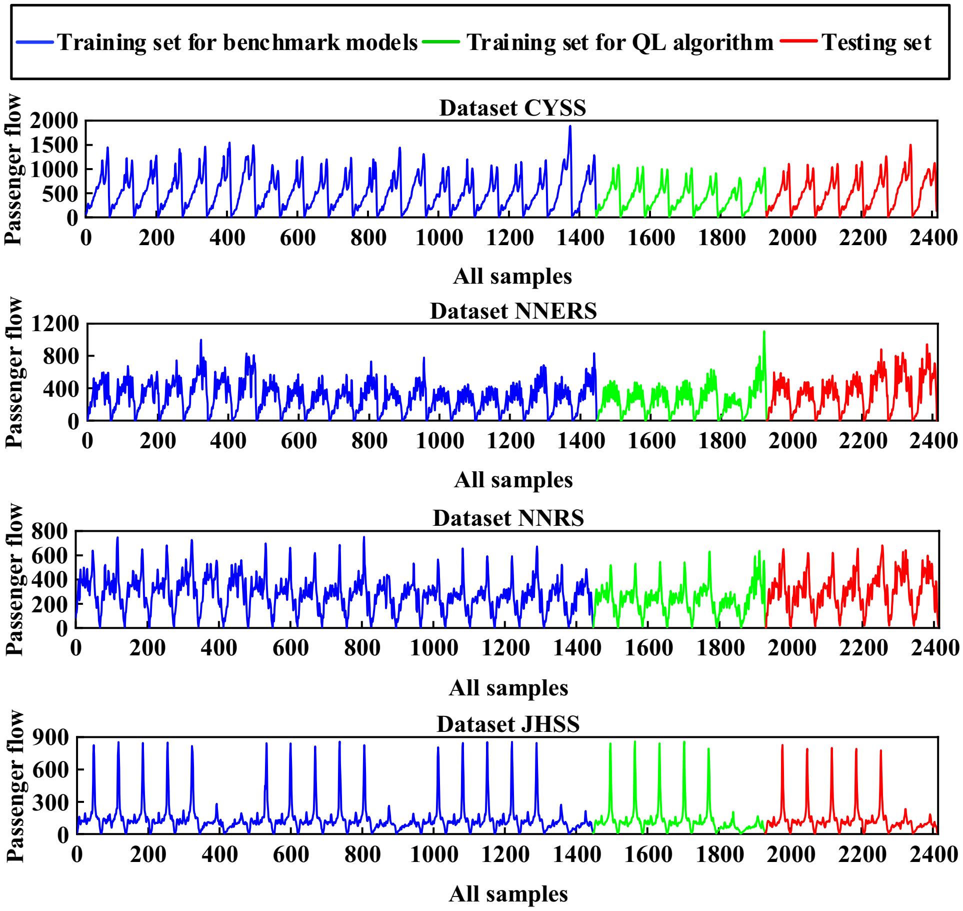


Fig. 3 Passenger flow data for four Nanning Metro stations

3.1 Dataset Description

The datasets were collected from the inbound ridership data of four URT stations in Nanning, including Chaoyang Square Station (CYSS), Nanning East Railway Station (NNERS), Nanning Railway Station (NNRS), and Jinhu Square Station (JHSS). CYSS is the hub station with enormous passenger flow. NNRS and JJHSS are transfer stations. NNERS is a high-speed railway station with greater passenger flow. The automatic fare collection (AFC) data from July 26 to August 29, 2021, was used for experiments in the study. Since the operation time of Nanning Metro Line 1 is 6:30–23:00, the ridership data from 6:30–23:30 is filtered and retained. The datasets were aggregated for 15 minutes, and each dataset time series has 2415 continuous data points. Table 1 shows various statistical features of the datasets. Figure 3 shows the original ridership time series required for the experiment. Table 1 and Fig. 3 indicate that the original passenger flow datasets have nonlinear and nonstationary characteristics. Among them, 1449 data points in the first 3 weeks were utilized to train the two benchmark models of BP and LSTM, 483 data points in the fourth week were used to train the QL algorithm, and the remaining data were used to test the prediction capability of OVMD-FDE-QL.

3.2 Baseline Models

To verify the effectiveness of the proposed model, OVMD-FDE-QL was compared with some traditional and hybrid methods in experiments as follows:

BP: As mentioned above, this is a traditional machine-learning model. The experiment uses BP with a three-layer architecture of an input layer, a hidden layer, and an output layer. The learning rate is set to 0.001, and the hidden layer size is 64.

Temporal convolutional network (TCN): The TCN increases the layer depth by dilated convolution, thus allowing an exponentially large receiving domain. Dilated convolution can correctly handle longer sequences in a non-recursive manner, which is conducive to parallel computing and alleviates the gradient explosion problem. The exponential strategy is used to increase the dilated factor in the experiment.

LSTM: As described above, the model is good at dealing with time series. LSTM contains two hidden layers, one of which has 128 neurons, and the other has 64 neurons. The batch size is set to 16, the time step is set to 5, the optimizer uses Adam, the number of iterations is set to 250, and the learning rate is set to 0.001.

BP-LSTM-QL: This model is established by integrating the BP and LSTM prediction results through QL.

LSTM-TCN-QL: This model is built by integrating the prediction results of the TCN and LSTM through QL.

OVMD-BP: The model decomposes the original passenger flow time series through OVMD and then directly uses BP to predict all sub-series. Instead of integrating all the predicted results through QL, the model uses direct superposition.

OVMD-LSTM: Like OVMD-BP, this model directly uses LSTM to predict all sub-series and also does not use QL to integrate the prediction results of all sub-series.

OVMD-FDE: After OVMD decomposition, this model uses FDE to divide the high-frequency and low-frequency components. LSTM predicts the high-frequency components, and BP predicts the low-frequency components. Finally, the model does not use QL to integrate the prediction results of all sub-series.

EMD-FDE-QL: This model uses EMD to decompose the original passenger flow time series, and the rest is consistent with OVMD-FDE-QL.

EEMD-FDE-QL: This model uses EEMD to decompose the original passenger flow time series, and the rest is consistent with OVMD-FDE-QL.

CEEMDAN-FDE-QL: This model uses CEEMDAN to decompose the original passenger flow time series, and the rest is consistent with OVMD-FDE-QL.

WD-FDE-QL: This model uses WD to decompose the original passenger flow time series, and the rest is consistent with OVMD-FDE-QL.

OVMD-FDE-GBDT: After using OVMD decomposition and FDE to divide the sub-series, this model uses the GBDT ensemble method to obtain the final prediction results.

OVMD-FDE-NSGA-II: This model uses the NSGA-II ensemble method to obtain the final prediction results.

OVMD-FDE-IAOA: The IAOA is used in the ensemble method to obtain the final prediction results.

OVMD-FDE-GWO: The GWO is used in the ensemble method to obtain the final prediction results.

3.3 Experimental Configuration

All the experiments in the study were carried out on a Linux server (CPU: i9-10900X, GPU: RTX 3090). TensorFlow and Keras developed the proposed model. In the process of the IAOA optimizing VMD, the search population is 30 in two-dimensional space, the search range of k is set to be between 1 and 20, and the search range of α is set to be between 1 and 5000. The maximum number of iterations is 20. In the OVMD-FDE-QL model, LSTM contains two hidden layers, one with 128 neurons and the other with 64 neurons. The batch size is 16, the time step is 5, and the prediction step is 3. The optimizer uses Adam, the number of iterations is 250, and the learning rate is 0.001. The reward discount coefficient γ of the QL algorithm is 0.99, the learning rate α is 0.01, the initial greedy parameter ε is 1, the final greedy parameter is 0.1, and the maximum episode is 100. For the passenger flow prediction results, mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and standard deviation of the error (SDE) [60] were used to verify the accuracy of the prediction and estimate the prediction ability of the proposed hybrid model. The mathematical expression of these indicators can be written as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|\hat{y}_t - y_t\right| \tag{27}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{\hat{y}_t - y_t}{y_t}\right| \tag{28}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2} \tag{29}$$

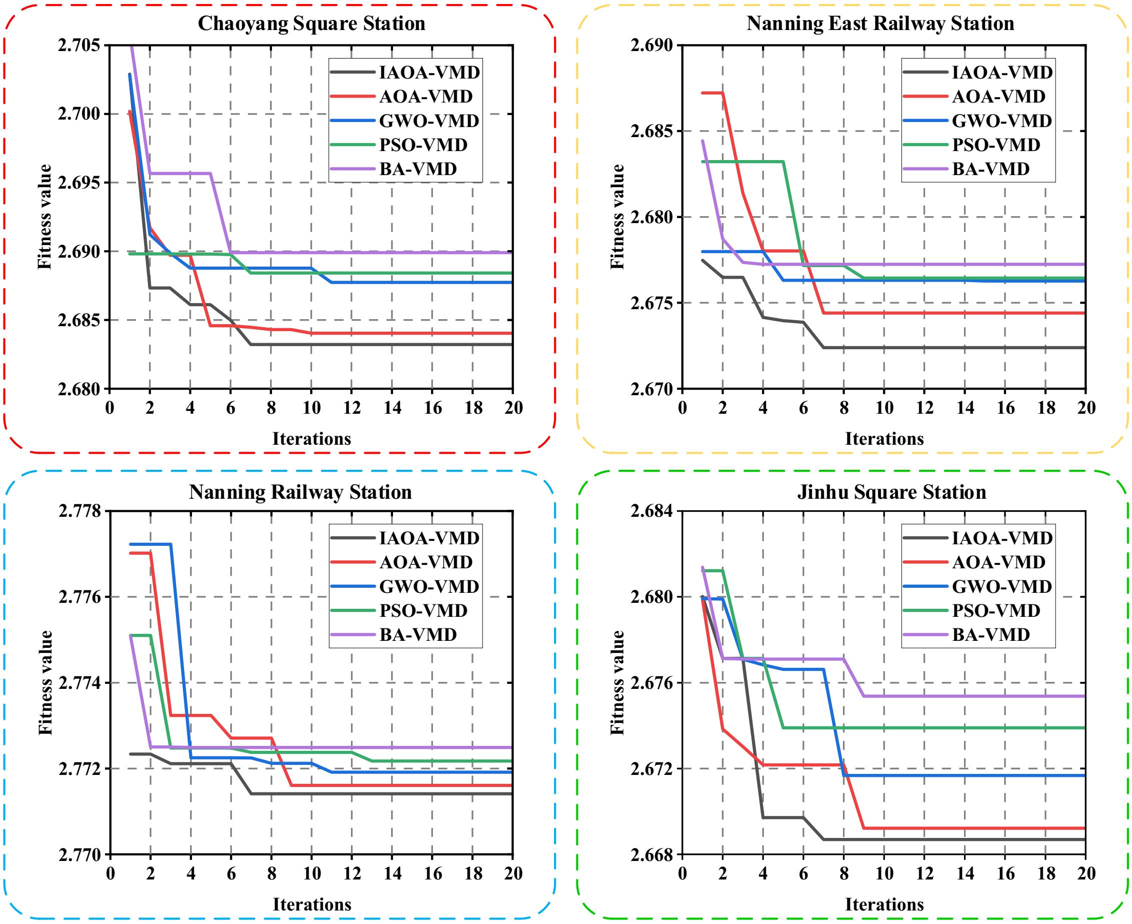


Fig. 4 Comparison of algorithm results

$$\mathrm{SDE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left[y_t - \hat{y}_t - \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)\right]^2} \tag{30}$$

3.4 Hyperparameter Analysis

3.4.1 VMD Parameter Optimization Results

To obtain the best modal component k and penalty factor α of VMD, the IAOA is used to optimize VMD with the minimum envelope entropy of the modal component as the fitness function. In addition, to prove the optimization ability of the IAOA, AOA, GWO, PSO, and BA [61] selected for optimization comparison, their optimization processes are shown in Fig. 4. It can be seen from Fig. 4 that the IAOA has achieved the smallest minimum value compared to other optimization algorithms, and the IAOA has the fastest convergence speed. This proves that the proposed IAOA has excellent performance in optimizing VMD. According to the optimization results of the IAOA, the optimal k and α are [15, 244] in CYSS, [19, 206] in NNERS, [19] in NNRS, and [3, 18] in JHSS.

3.4.2 OVMD Decomposition Results and FDE Calculation Results

After obtaining the best VMD parameters, the OVMD is applied to decompose the passenger flow data. Figure 5 shows the decomposition results of four ridership time series. The sub-series are more stable and regular than the original ridership data, which undoubtedly reduces the prediction pressure of the model.

OVMD obtains a finite number of IMF components of different complexity by decomposition. The calculation error will increase if a single prediction model is used to predict each sub-component directly. To reduce the calculation error, the complexity of each IMF sub-series is calculated by FDE and divided into high-frequency IMF and low-frequency IMF. Furthermore, the FDE threshold is set to 0.8, the IMF components below 0.8 are the low-frequency

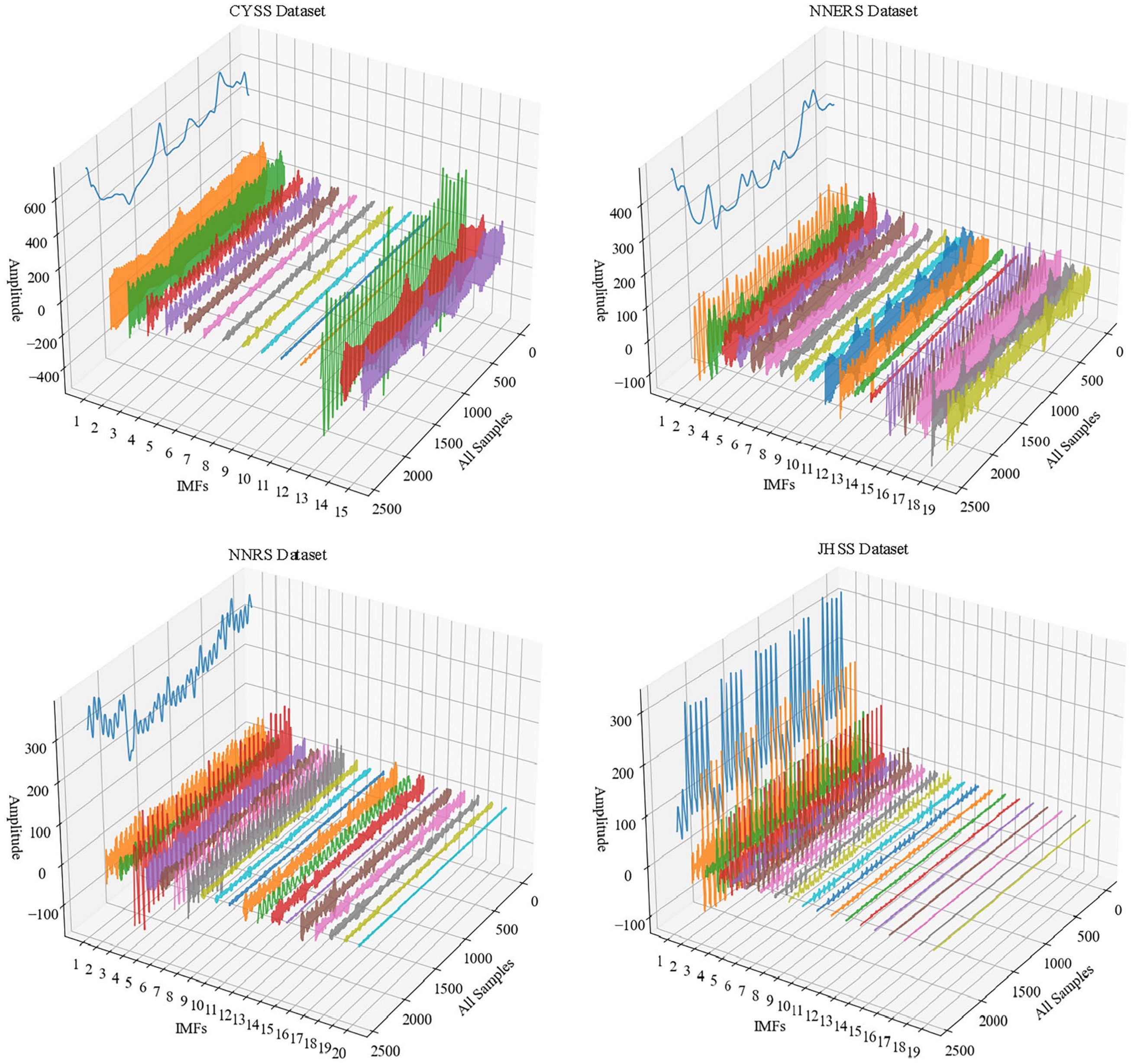


Fig. 5 Decomposition results of OVMD on four station datasets

series, and the high-frequency series is higher than 0.8 [37]. The FDE calculation results of IMF components in the four datasets are shown in Fig. 6, which reveals that the first component in the four datasets is the low-frequency component, and the rest are the high-frequency components. Therefore, BP is used to predict IMF1, and LSTM is used to predict the remaining IMFs.

3.4.3 FDE Threshold Selection

When FDE is used as a criterion for the complexity of each IMF, selecting the FDE threshold is the key to determine the combined forecasting model. If the FDE threshold is too high, BP will predict some high-frequency IMFs. On the contrary, if the FDE threshold is too low, LSTM will predict some low-frequency IMFs. The combined prediction model can obtain higher accuracy by selecting the appropriate FDE threshold. Therefore, different values were selected in [0.1, 0.2, 0.5, 0.8, 1, 2, 3, 4, 4.5] to analyze the sensitivity of the FDE threshold. When the FDE of IMF is less than the threshold, BP is used to predict the components; otherwise, LSTM is used to predict the components. Finally, the QL algorithm is used to integrate the prediction results of each component. The calculation results are shown in Fig. 7.

Theoretically, when the threshold for FDE is 0.1, all IMFs use LSTM for prediction. When the threshold for

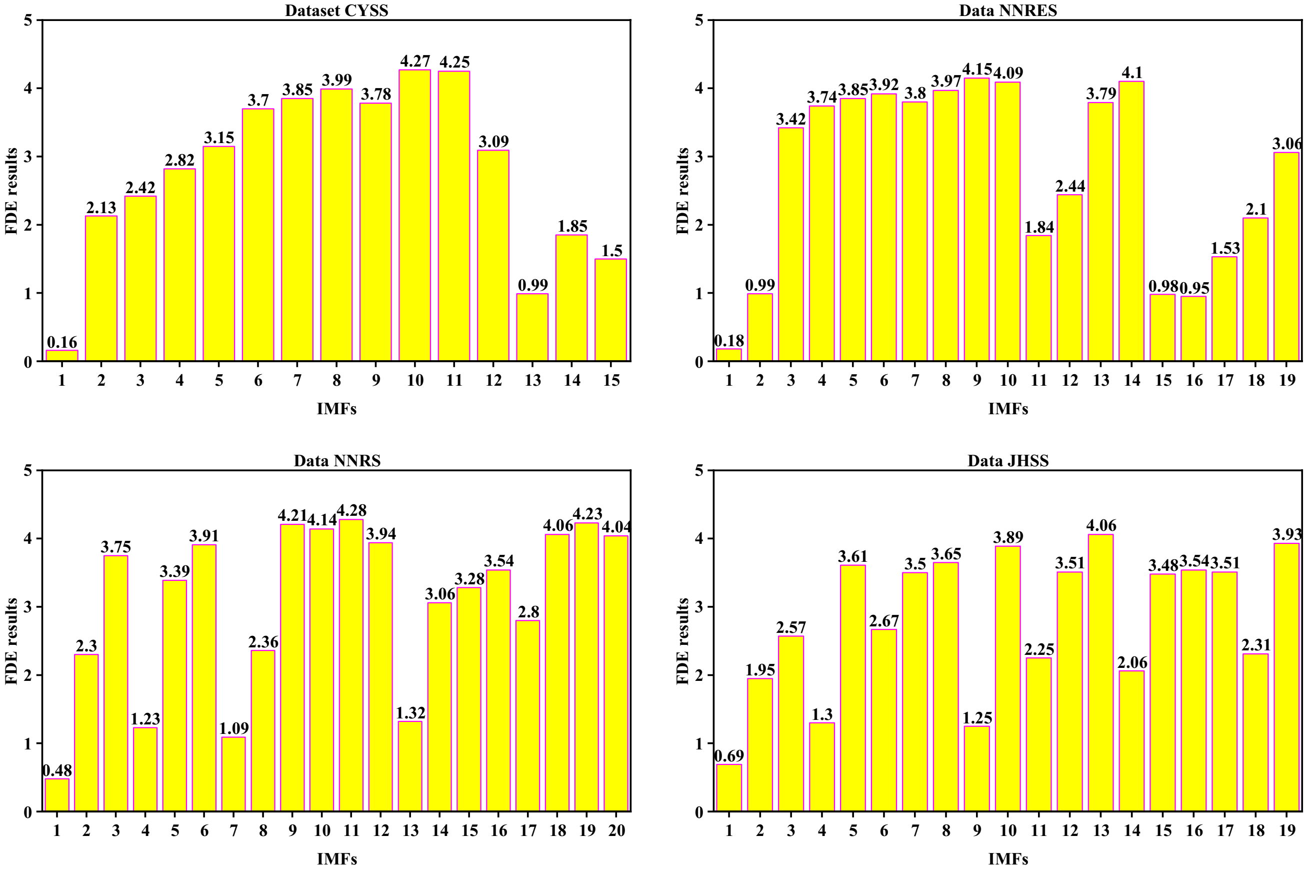


Fig. 6 The FDE values of each sub-series after OVMD decomposition of four passenger flow datasets

FDE is 4.5, all IMFs use BP for prediction. As the threshold of FDE increases from 0.1, more and more low-frequency components begin to choose BP for prediction, so the prediction error of the reconstructed signal begins to decrease. When the threshold of FDE exceeds a specific value, high-frequency IMFs increasingly begin to choose BP for prediction, and the prediction error will increase. It can be seen from Fig. 7 that as the FDE threshold increases, the error of the reconstructed signal decreases first and then increases, which is in line with expectations. Combining the thresholds of the four cases in Fig. 7, the prediction error is the smallest when the threshold of FDE is 0.8. Therefore, the threshold of FDE is set to 0.8 in this paper. In general, when the threshold of FDE is selected appropriately, all low-frequency IMFs are predicted by BP, and LSTM predicts high-frequency IMFs. The prediction error of the hybrid prediction model for the reconstructed signal is smaller than that of the single model. This reflects the advantages of the hybrid model over the single model and proves that FDE can well reflect the complexity of the time series signal. Using FDE to divide IMF into high-frequency and low-frequency components is reasonable.

3.5 Comparative Experiments

3.5.1 Comparison of Single and Integrated Models Without Decomposition

This section compares the proposed OVMD-FDE-QL model with several single and integrated models' prediction results. The single models include BP, LSTM, and the temporal convolutional network (TCN). The integrated models include a hybrid model of BP and LSTM integrated by QL and a hybrid model of LSTM and TCN integrated by QL. The results on the four datasets are shown in Table 2, where the best prediction results are bolded in the table, and the sub-optimal results are shown next to *.

The prediction results in Table 2 show that the prediction ability of OVMD-FDE-QL is more evident than that of other prediction models. OVMD data do not preprocess these models but are directly predicted. Single and integrated models cannot predict the nonlinear passenger flow time series well. Therefore, it is wise to process ridership using the decomposition method for URT station passenger flow prediction. In a one-step prediction in CYSS, the MAE of OVMD-FDE-QL is 2.695. Compared with LSTM-TCN-QL,

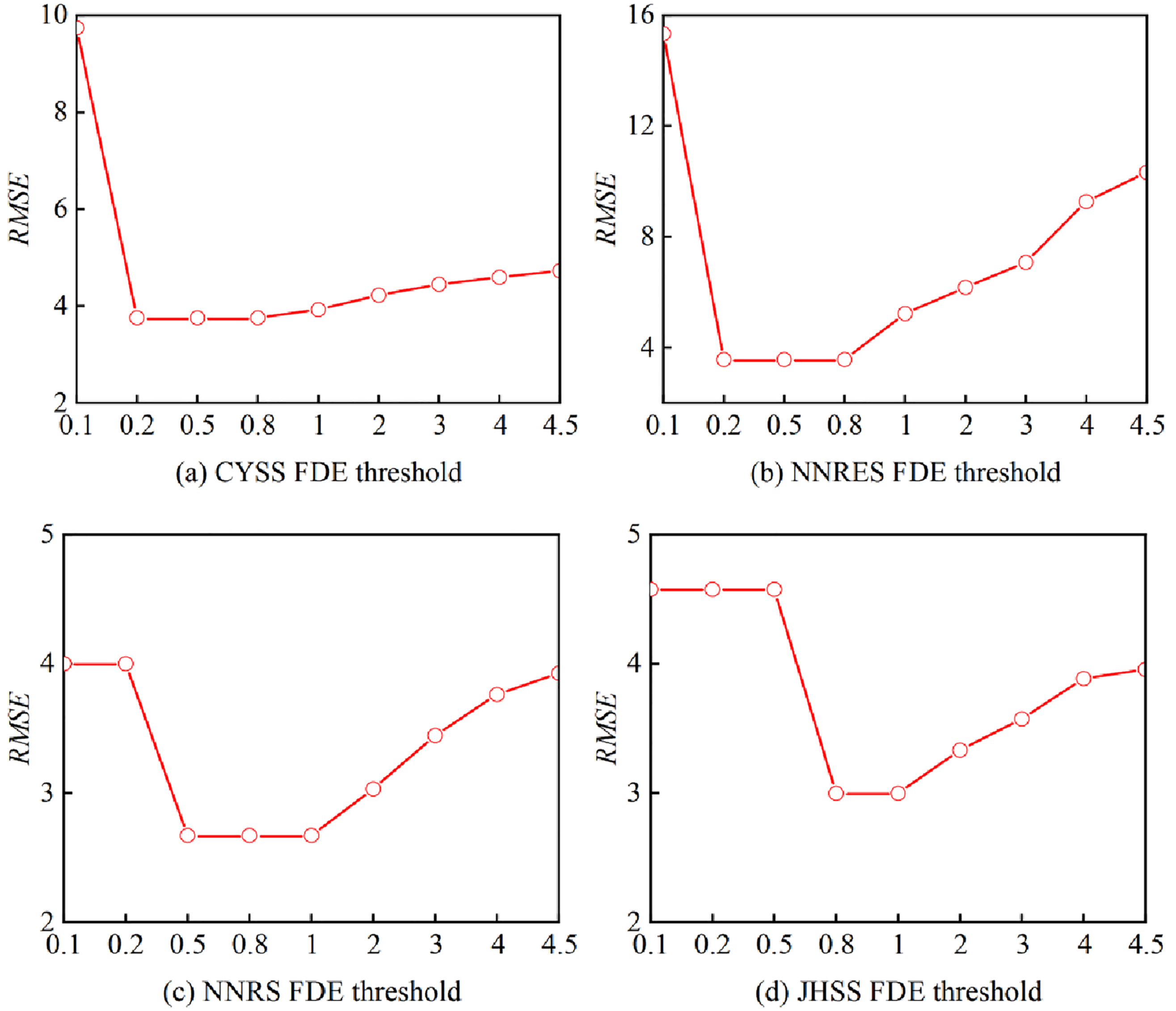


Fig. 7 The relationship between the FDE threshold and the prediction error

BP-LSTM-QL, LSTM, TCN, and BP, the prediction accuracy of OVMD-FDE-QL is improved by 85.21%, 85.54%, 86.36%, 89.42%, and 89.95%, respectively. In a one-step prediction in NNERS, the MAPE of OVMD-FDE-QL is 0.198. Compared with LSTM-TCN-QL, BP-LSTM-QL, LSTM, TCN, and BP, the prediction accuracy of OVMD-FDE-QL is improved by 82.37%, 82.26%, 85.41%, 82.71%, and 82.89%, respectively. In a one-step prediction in NNRS, the RMSE of OVMD-FDE-QL is 2.673. Compared with LSTM-TCN-QL, BP-LSTM-QL, LSTM, TCN, and BP, the prediction accuracy of OVMD-FDE-QL is improved by 91.25%, 90.94%, 91.62%, 92.64%, and 92.3%, respectively. In a one-step prediction in JHSS, the SDE of OVMD-FDE-QL is 2.985. Compared with LSTM-TCN-QL, BP-LSTM-QL, LSTM, TCN, and BP, the prediction accuracy of OVMD-FDE-QL is improved by 80.15%, 78.99%, 81.09%, 84.24%, and 81.06%, respectively.

As a shallow neural network, the nonlinear prediction ability of BP is weaker than that of TCN and LSTM deep learning models. The TCN utilizes dilated causal convolutions to provide the model with sequential logic without increasing the parameter size, and it achieves associations between any two temporal nodes through stacking. TCN may need to be improved when dealing with particular sequence data, such as data with nonstationary, nonlinear relationships or very long sequences. In these cases, LSTM can better adapt to the characteristics of the data and provide better prediction performance than TCN. For example, in a two-step

Table 2 Results of OVMD-FDE-QL and baseline models

Dataset	Model	One-step				Two-step				Three-step			
		MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE
CYSS	**OVMD-FDE-QL**	**2.975**	**0.075**	**3.759**	**3.405**	**3.456**	**0.042**	**4.345**	**4.319**	**6.231**	**0.071**	**7.954**	**7.031**
	LSTM-TCN-QL*	20.121	0.624	27.706	27.291	45.457	0.634	69.253	68.369	69.342	1.288	107.402	106.106
	BP-LSTM-QL	20.578	0.634	28.058	27.719	46.003	0.766	70.073	69.245	69.945	1.392	107.672	106.289
	LSTM	21.817	0.709	30.669	30.373	47.697	0.791	71.563	71.088	70.659	1.503	110.085	108.645
	TCN	28.117	0.752	38.963	37.478	53.004	0.835	79.261	78.066	76.196	1.597	119.941	118.766
	BP	29.606	0.793	39.879	38.107	59.355	0.867	89.648	89.633	83.768	2.796	130.269	130.259
NNERS	**OVMD-FDE-QL**	**2.695**	**0.198**	**3.57**	**3.564**	**5.439**	**0.242**	**6.931**	**6.93**	**10.44**	**0.411**	**13.039**	**13.037**
	LSTM-TCN-QL	28.612	1.123	43.048	43.046	66.699	4.059	99.41	99.095	90.222	6.624	130.334	129.742
	BP-LSTM-QL*	26.744	1.116	39.78	39.766	65.129	4.016	95.289	95.046	88.865	6.543	125.343	124.878
	LSTM	30.452	1.357	47.282	44.405	68.92	4.078	102.236	100.375	95.101	6.026	134.386	131.111
	TCN	34.185	1.145	49.641	46.436	75.317	4.504	104.727	101.716	93.899	7.172	135.96	132.542
	BP	35.756	1.157	50.243	49.687	76.776	5.773	107.588	103.678	99.269	9.455	138.153	138.558
NNRS	**OVMD-FDE-QL**	**2.136**	**0.015**	**2.673**	**2.669**	**3.961**	**0.024**	**4.854**	**4.854**	**5.818**	**0.036**	**7.364**	**7.162**
	LSTM-TCN-QL*	22.698	0.127	30.535	30.465	53.797	0.28	74.71	74.475	74.513	0.493	99.358	98.788
	BP-LSTM-QL	22.643	0.13	29.499	29.466	54.156	0.303	72.813	72.623	75.053	0.492	97.555	97.03
	LSTM	23.983	0.139	31.967	31.847	55.126	0.277	75.954	75.626	75.065	0.43	100.494	99.517
	TCN	26.947	0.144	36.29	35.901	59.15	0.335	81.558	81.386	80.408	0.649	105.999	105.641
	BP	27.292	0.142	34.725	33.793	61.696	0.38	81.537	80.663	84.313	0.645	107.428	106.988
JHSS	**OVMD-FDE-QL**	**1.88**	**0.025**	**2.998**	**2.985**	**3.287**	**0.046**	**4.379**	**4.374**	**4.771**	**0.063**	**6.413**	**6.4**
	LSTM-TCN-QL	9.399	0.114	15.069	15.034	20.632	0.214	40.169	40.169	30.887	0.336	63.586	63.586
	BP-LSTM-QL*	8.619	0.101	14.221	14.209	21.043	0.228	39.45	39.45	32.233	0.373	63.784	63.784
	LSTM	10.159	0.133	15.823	15.785	21.615	0.238	40.582	40.57	32.565	0.377	64.354	64.344
	TCN	12.054	0.128	20.012	18.941	22.964	0.23	44.444	43.774	32.072	0.338	66.752	66.551
	BP	10.97	0.108	16.672	15.758	23.657	0.233	41.873	41.339	37.814	0.417	67.676	67.55

prediction in NNERS, the MAE of LSTM is 68.92, and TCN and BP correspond to 75.317 and 76.776, respectively. LSTM's MAE decreased by 8.49% and 10.23%, respectively.

In addition, the results of LSTM-TCN-QL and BP-LSTM-QL are better than those of BP, LSTM, and TCN. This shows that the QL algorithm can integrate the two benchmark models well and obtain better prediction performance than the benchmark models. For example, in a three-step prediction in NNRS, the MAE of LSTM-TCN-QL is 74.513. Compared with LSTM, TCN, and BP, the predictive performance of the LSTM-TCN-QL model improved by 0.74%, 7.33%, and 11.62%, respectively. Although LSTM-TCN-QL and BP-LSTM-QL improve prediction accuracy to a certain extent, they still need to be preprocessed and have a large gap compared to OVMD-FDE-QL, which OVMD decomposes. The hybrid model of data preprocessing technology based on OVMD has better predictive ability than the single models. Data preprocessing techniques can help reduce the impact of noise in the data, thereby significantly improving prediction performance. Figure 8 shows the predicted value, error value, scatter plot, and evaluation index comparison of each model on CYSS.

3.5.2 Comparison with Other OVMD-Based Models

This section combines OVMD with different prediction models to compare the prediction performance with OVMD-FDE-QL. In the experiment, OVMD-BP and OVMD-LSTM models are designed. These two models do not calculate the complexity of IMFs but directly use BP and LSTM for prediction. Moreover, the OVMD-FDE model is constructed, which obtains the final prediction result by superimposing all the results of the sub-series and is shown in Table 3.

The results of OVMD-BP are better than those of OVMD-LSTM in the four datasets, while BP predicted worse than LSTM in experiment 3.5.1. This is because OVMD-BP and OVMD-LSTM do not distinguish between high-frequency and low-frequency series by FDE but directly use a model to make predictions. The low-frequency components contain more inherent nature and information about passenger flow. Accurately grasping

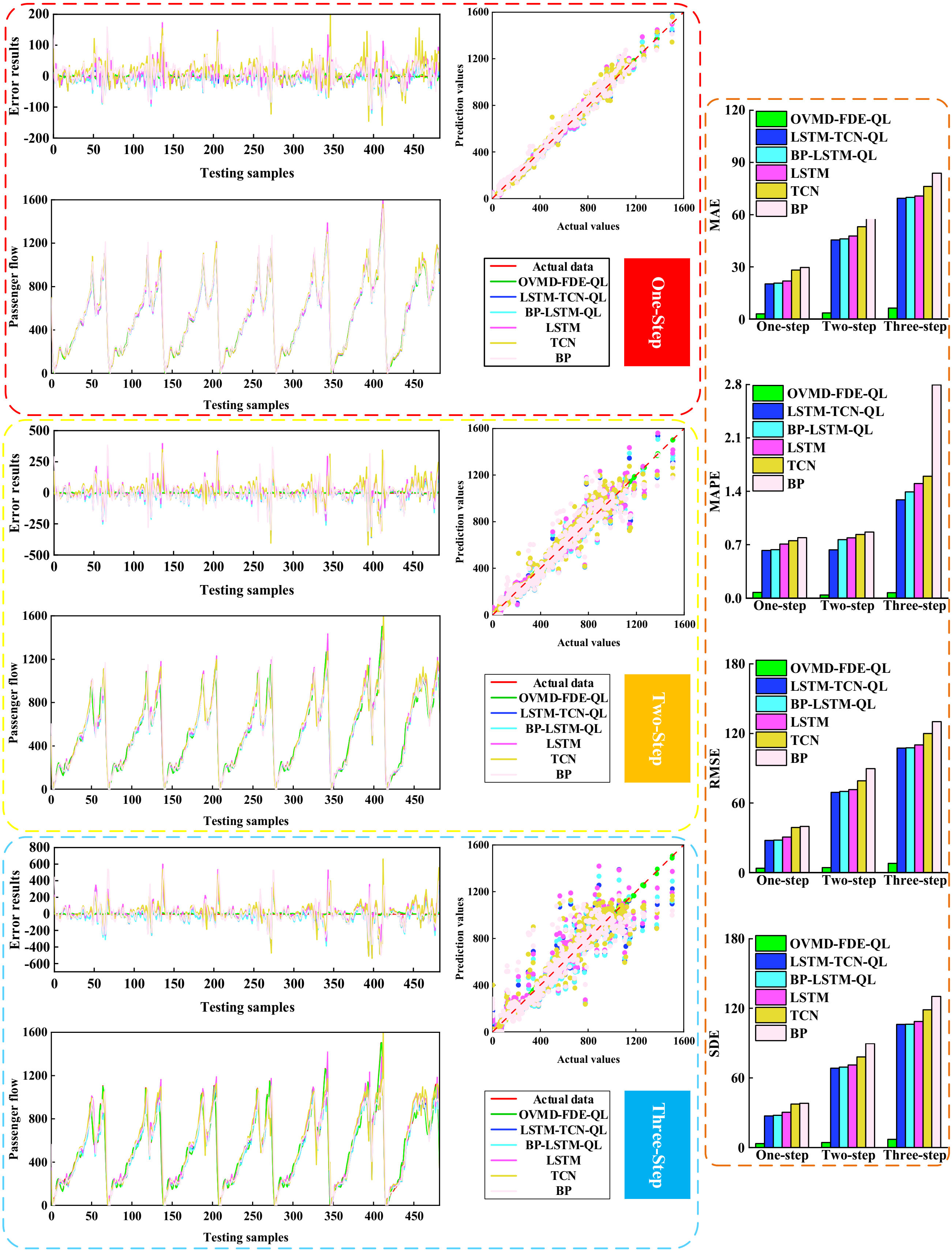


Fig. 8 Results for each model in CYSS

Table 3 Results based on OVMD combination models

Dataset	Model	One-step				Two-step				Three-step			
		MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE
CYSS	**OVMD-FDE-QL**	**2.975**	**0.075**	**3.759**	**3.405**	**3.456**	**0.042**	**4.345**	**4.319**	**6.231**	**0.071**	**7.954**	**7.031**
	OVMD-FDE*	3.037	0.115	3.825	3.754	4.797	0.055	5.899	5.156	7.094	0.107	8.921	8.906
	OVMD-BP	4.076	0.124	4.999	4.863	5.419	0.066	6.653	6.644	9.642	0.139	11.649	11.591
	OVMD-LSTM	6.028	0.16	11.038	10.416	7.236	0.069	12.522	11.21	10.262	0.162	14.28	13.844
NNERS	**OVMD-FDE-QL**	**2.695**	**0.198**	**3.57**	**3.564**	**5.439**	**0.242**	**6.931**	**6.93**	**10.44**	**0.411**	**13.039**	**13.037**
	OVMD-FDE*	2.73	0.21	3.697	3.678	5.655	0.268	7.24	7.228	11.207	0.477	14.351	14.35
	OVMD-BP	6.843	0.359	11.946	11.915	7.356	0.392	12.334	12.304	11.28	0.529	14.841	14.822
	OVMD-LSTM	13.699	0.424	26.285	24.751	14.248	0.479	26.542	25.516	16.06	0.572	27.739	26.867
NNRS	**OVMD-FDE-QL**	**2.136**	**0.015**	**2.673**	**2.669**	**3.961**	**0.024**	**4.854**	**4.854**	**5.818**	**0.036**	**7.364**	**7.162**
	OVMD-FDE*	2.913	0.017	3.828	2.816	4.029	0.025	5.107	5.026	6.4	0.039	7.931	7.926
	OVMD-BP	3.52	0.022	4.392	3.632	5.183	0.032	6.706	6.704	7.346	0.048	9.416	9.399
	OVMD-LSTM	3.531	0.036	4.902	4.892	6.756	0.039	8.277	7.404	10.083	0.054	12.234	11.078
JHSS	**OVMD-FDE-QL**	**1.88**	**0.025**	**2.998**	**2.985**	**3.287**	**0.046**	**4.379**	**4.374**	**4.771**	**0.063**	**6.413**	**6.4**
	OVMD-FDE*	2.866	0.033	4.604	4.209	3.95	0.059	6.397	6.019	5.582	0.08	8.975	8.495
	OVMD-BP	3.522	0.034	4.875	4.301	3.951	0.062	6.447	6.024	5.782	0.077	9.276	8.904
	OVMD-LSTM	3.772	0.039	5.558	4.758	8.301	0.075	12.197	10.605	11.809	0.144	18.136	16.918

the inherent nature of short-term passenger flow is the key to improving prediction performance. However, LSTM produces an overfitting phenomenon when predicting stationary low-frequency series, which influences the final prediction results. For example, in a one-step prediction in CYSS, the MAPE of OVMD-BP is 0.124, and OVMD-LSTM corresponds to 0.16. The prediction accuracy of OVMD-BP has improved by 22.5%. OVMD-FDE uses FDE to divide sub-series, maximizing the strengths of each model and improving prediction accuracy. In a two-step prediction in NNERS, the MAPE of OVMD-FDE is 0.268. OVMD-BP and OVMD-LSTM correspond to 0.392 and 0.479, respectively. The predictive performance of the OVMD-FDE model improved by 31.75% and 44.15% through FDE. All of these demonstrate the importance of dividing sub-series.

Moreover, by comparing OVMD-FDE-QL and OVMD-FDE, it can be found that the QL algorithm can effectively integrate BP and LSTM models, further improving the prediction accuracy. In a three-step prediction in NNRS, compared with the OVMD-FDE, the prediction accuracy of the OVMD-FDE-QL has improved by 7.69%. In a three-step prediction in JHSS, compared with the OVMD-FDE, the prediction accuracy of the OVMD-FDE-QL has improved by 21.25%. Figure 9 compares each model's predicted value, error, scatter plot, and evaluation index on NNERS. The experiment in this part shows that analyzing the complexity of sub-series is very important when using data decomposition technology for prediction. In addition, the QL ensemble method can effectively improve prediction accuracy.

3.5.3 *Comparison of Integrated Prediction Models Constructed Using Various Decomposition Algorithms*

This section compares the proposed model to EMD-based, EEMD-based, CEEMDAN-based, and WD-based models. The experimental results are shown in Table 4. The results of Table 4 show that EMD has a good processing effect on nonlinear and nonstationary signals, but the inherent defects of mode mixing limit its wide application. EEMD solves mode mixing by adding white noise. Based on this, CEEMDAN adds adaptive white noise to each decomposition to improve the integrity of EEMD and reduce the reconstruction error. The basic principles of these three algorithms are the same, but the noise is different. Thus, their prediction performance is different. For example, in a one-step prediction in CYSS, the RMSE values of EMD-FDE-QL, EEMD-FDE-QL, and CEEMDAN-FDE-QL are 12.826, 11.029, and 12.464, respectively. The predictive abilities of CEEMDAN-FDE-QL and EMD-FDE-QL are comparable. The capability of EEMD-FDE-QL is better than that of CEEMDAN-FDE-QL and EMD-FDE-QL. The decomposition stability of WD is better than that of EMD, so the capability of WD-FDE-QL is better than that of all EMD-based hybrid models. For example, in a two-step prediction in NNERS, compared to EEMD-FDE-QL, the prediction accuracy of WD-FDE-QL has improved by 2.81%.

In contrast, this paper uses OVMD for data preprocessing, which gives the proposed OVMD-FDE-QL model the best prediction accuracy. In a three-step prediction in CYSS, compared with the WD-FDE-QL, the prediction accuracy of the

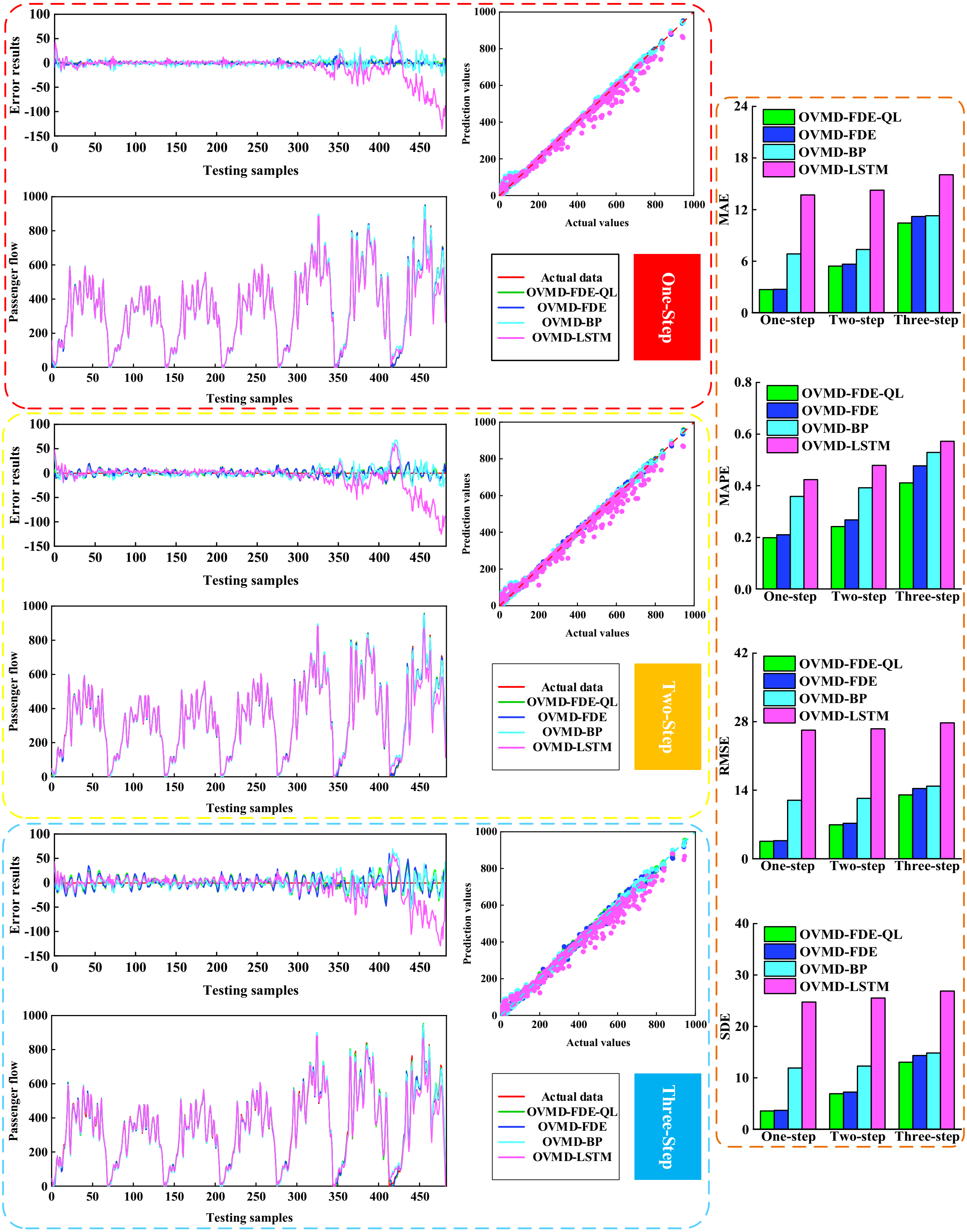


Fig. 9 Results of each model in NNERS

Table 4 Comparison of prediction performance of models based on different decomposition techniques

Dataset	Model	One-step				Two-step				Three-step			
		MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE
CYSS	**OVMD-FDE-QL**	**2.975**	**0.075**	**3.759**	**3.405**	**3.456**	**0.042**	**4.345**	**4.319**	**6.231**	**0.071**	**7.954**	**7.031**
	WD-FDE-QL*	7.121	0.104	9.589	9.585	13.624	0.311	18.283	18.215	17.134	0.288	22.556	22.321
	CEEMDAN-FDE-QL	12.464	0.147	18.724	18.724	25.874	0.491	37.658	37.656	36.716	0.811	51.876	51.872
	EEMD-FDE-QL	11.029	0.161	14.43	14.428	18.538	0.299	25.782	25.777	28.709	0.454	41.973	41.967
	EMD-FDE-QL	12.826	0.149	17.871	17.866	24.33	0.494	33.367	33.344	34.314	1.001	45.999	45.971
NNERS	**OVMD-FDE-QL**	**2.695**	**0.198**	**3.57**	**3.564**	**5.439**	**0.242**	**6.931**	**6.93**	**10.44**	**0.411**	**13.039**	**13.037**
	WD-FDE-QL*	13.467	0.437	20.811	20.809	19.336	0.63	28.638	28.629	19.922	0.787	30.572	30.539
	CEEMDAN-FDE-QL	17.829	0.726	24.338	24.328	33.006	1.085	45.648	45.646	41.559	1.61	57.099	57.099
	EEMD-FDE-QL	14.146	0.499	21.485	21.479	20.658	0.831	29.466	29.46	31.826	0.99	43.987	43.979
	EMD-FDE-QL	17.89	0.744	24.952	24.947	31.983	0.978	45.193	45.188	38.212	1.133	52.613	52.607
NNRS	**OVMD-FDE-QL**	**2.136**	**0.015**	**2.673**	**2.669**	**3.961**	**0.024**	**4.854**	**4.854**	**5.818**	**0.036**	**7.364**	**7.162**
	WD-FDE-QL*	7.329	0.041	9.695	9.695	14.045	0.098	18.037	18.033	13.282	0.099	16.768	16.737
	CEEMDAN-FDE-QL	15.356	0.01	22.521	22.516	32.352	0.203	46.855	46.853	39.988	0.237	54.68	54.681
	EEMD-FDE-QL	10.049	0.082	13.893	13.89	15.986	0.113	21.453	21.453	27.627	0.168	37.11	37.109
	EMD-FDE-QL	17.714	0.091	27.107	27.09	33.847	0.181	49.625	49.619	42.58	0.268	58.814	58.807
JHSS	**OVMD-FDE-QL**	**1.88**	**0.025**	**2.998**	**2.985**	**3.287**	**0.046**	**4.379**	**4.374**	**4.771**	**0.063**	**6.413**	**6.4**
	WD-FDE-QL*	3.699	0.044	5.903	5.888	7.501	0.093	11.285	11.285	8.403	0.12	11.826	11.824
	CEEMDAN-FDE-QL	8.672	0.087	14.72	14.719	17.497	0.177	32.637	32.629	21.809	0.226	37.704	37.69
	EEMD-FDE-QL	5.379	0.077	8.545	8.545	7.764	0.105	12.103	12.102	11.946	0.142	18.555	18.554
	EMD-FDE-QL	8.119	0.095	13.598	13.597	18.296	0.193	33.374	33.374	23.212	0.265	38.048	38.047

OVMD-FDE-QL has improved by 75.35%. In a three-step prediction in NNERS, compared with the WD-FDE-QL, the prediction accuracy of the OVMD-FDE-QL has improved by 47.78%. In a three-step prediction in NNRS, compared with the WD-FDE-QL, the prediction accuracy of the OVMD-FDE-QL has improved by 63.64%. In a three-step prediction in JHSS, compared with the WD-FDE-QL, the prediction accuracy of the OVMD-FDE-QL has improved by 47.5%. Figure 10 compares the predicted value, error, scatter plot, and evaluation index for each model on the NNRS. It can also be seen from Fig. 10 that the hybrid strategy based on OVMD is consistently ranked higher than the hybrid strategy based on other decomposition algorithms and has a very stable prediction ability.

3.5.4 Comparison with Different Ensemble Learning Strategy Models

This section aims to use different ensemble strategies to compare the capabilities of each model. In the experiment, the gradient boosting decision tree (GBDT), the non-dominated sorting genetic algorithm II (NAGA-II) [62], the IAOA, and the GWO algorithm are selected for comparison. The experimental results are shown in Table 5.

It can be seen from Table 5 that the QL ensemble demonstrates the best capability, followed by the ensemble strategy based on GBDT. GBDT is a stacking-based ensemble strategy. Although the GBDT ensemble strategy is second only to the QL algorithm, it is not easier to understand than the weighted ensemble. NSGA-II is a multi-objective optimization algorithm, while IAOA and GWO are single-objective optimization algorithms. There is no doubt that the capability of a multi-objective optimization algorithm is better than that of a single-objective optimization algorithm. Thus, NSGA-II is superior to IAOA and GWO in performance. Since the optimization performance of GWO is not as good as that of IAOA, the ensemble strategy based on GWO is better than that based on IAOA. Unlike the meta-heuristic optimization algorithm, the QL algorithm has achieved excellent results in weight optimization through reward and punishment mechanisms. In the three-step prediction of the four datasets, compared with the GBDT-based ensemble strategy, the QL-based ensemble improves the prediction results of the model by 42.82%, 19.82%, 2.93%, and 6.21%, respectively. Compared with the multi-objective optimization algorithm, the QL improves the model's prediction performance by 45.14%, 23.6%, 11.44%, and 13.74%, respectively, and compared with the single-objective optimization algorithm, the QL improves the model's prediction performance by 72.49%, 25.44%, 13.74%, and 19.03%, respectively. Figure 11 compares the predicted value, error, scatter plot, and evaluation index of each model for experiment 4 on the NNRS. All in all, the ensemble strategy based on QL is very effective.

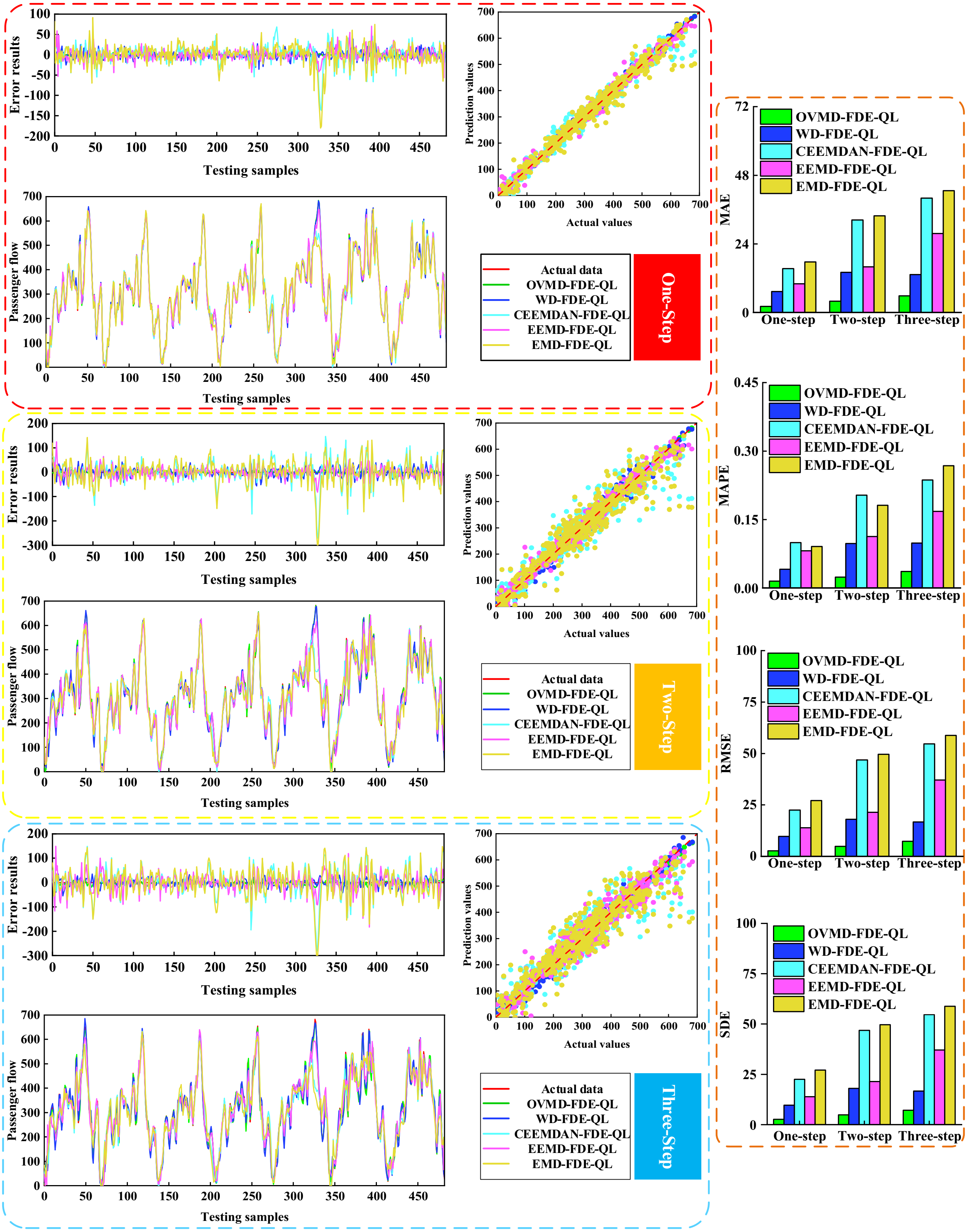


Fig. 10 Results of different data preprocessing techniques in NNRS

Table 5 Comparison of predictive capability of hybrid models with different ensemble strategies

Dataset	Model	One-step				Two-step				Three-step			
		MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE
CYSS	**OVMD-FDE-QL**	**2.975**	**0.075**	**3.759**	**3.405**	**3.456**	**0.042**	**4.345**	**4.319**	**6.231**	**0.071**	**7.954**	**7.031**
	OVMD-FDE-GBDT*	3.179	0.094	3.955	3.947	3.6	0.075	4.557	4.534	6.52	0.123	8.143	8.224
	OVMD-FDE-NSGA-II	3.476	0.1	4.395	4.237	4.273	0.079	5.447	4.556	6.541	0.129	8.243	8.229
	OVMD-FDE-IAOA	5.679	0.119	7.482	7.479	7.161	0.252	9.412	9.528	8.025	0.256	10.243	10.225
	OVMD-FDE-GWO	5.919	0.126	7.649	7.646	7.271	0.26	9.544	9.593	10.021	0.442	13.274	13.251
NNERS	**OVMD-FDE-QL**	**2.695**	**0.198**	**3.57**	**3.564**	**5.439**	**0.242**	**6.931**	**6.93**	**10.44**	**0.411**	**13.039**	**13.037**
	OVMD-FDE-GBDT*	2.643	0.207	3.614	3.605	5.579	0.255	7.105	7.083	10.46	0.513	13.142	13.147
	OVMD-FDE-NSGA-II	2.702	0.212	3.654	3.636	5.608	0.261	7.139	7.131	10.481	0.538	13.218	13.217
	OVMD-FDE-IAOA	3.23	0.242	4.558	4.551	5.849	0.274	7.64	7.638	11.132	0.551	14.079	14.072
	OVMD-FDE-GWO	5.546	0.298	7.036	7.032	7.105	0.324	8.893	8.885	11.382	0.688	14.606	14.605
NNRS	**OVMD-FDE-QL**	**2.136**	**0.015**	**2.673**	**2.669**	**3.961**	**0.024**	**4.854**	**4.854**	**5.818**	**0.036**	**7.364**	**7.162**
	OVMD-FDE-GBDT*	2.543	0.019	3.252	3.17	4.455	0.026	5.558	5.257	6.851	0.038	8.329	8.209
	OVMD-FDE-NSGA-II	3.103	0.024	3.875	3.247	4.886	0.029	6.045	5.56	7.18	0.041	9.197	8.314
	OVMD-FDE-IAOA	3.337	0.028	4.007	3.873	5.018	0.031	6.125	6.045	7.461	0.042	9.218	9.205
	OVMD-FDE-GWO	4.354	0.028	5.516	5.5	5.208	0.032	6.456	6.455	7.645	0.045	9.544	9.542
JHSS	**OVMD-FDE-QL**	**1.88**	**0.025**	**2.998**	**2.985**	**3.287**	**0.046**	**4.379**	**4.374**	**4.771**	**0.063**	**6.413**	**6.4**
	OVMD-FDE-GBDT*	2.01	0.028	3.133	3.088	3.388	0.048	4.692	4.692	4.899	0.068	6.565	6.558
	OVMD-FDE-NSGA-II	2.022	0.028	3.184	3.175	3.451	0.049	4.846	4.825	5.103	0.074	6.818	6.793
	OVMD-FDE-IAOA	2.474	0.038	3.592	3.288	3.584	0.051	4.915	4.845	5.235	0.078	7.159	6.902
	OVMD-FDE-GWO	2.584	0.053	3.725	3.725	3.666	0.072	5.161	4.907	5.369	0.099	7.318	7.313

3.6 Ablation Experiments

To explore the influence of different components on the performance of OVMD-FDE-QL, ablation experiments were carried out in this part, and the experimental results were analyzed. The designed variant models are as follows:

OVMD-BP-QL: This model removes the FDE algorithm, directly uses BP to predict all components after OVMD decomposition, and uses QL to ensemble the prediction results of all components.

OVMD-LSTM-Q: This model also removes the FDE algorithm, directly uses LSTM to predict all components after OVMD decomposition, and uses QL to ensemble the prediction results of all components.

VMD-FDE-Q: This model does not use IAOA to find the best parameters of VMD but directly selects the parameters of VMD through experience, and the remaining components are consistent with OVMD-FDE-QL. In addition, the number of modal components decomposed by VMD is consistent with EMD.

OVMD-FD: This model is the comparative baseline model in the above experiment, which has been described in detail above.

All the parameters of these variant models in the experiment are the same as those of OVMD-FDE-QL. The passenger flow prediction results at the four subway stations are shown in Table 6. By comparing OVMD-FDE-QL, OVMD-BP-QL, and OVMD-LSTM-QL, it can be seen that FDE is an essential part of the hybrid model. Through FDE, the most suitable prediction model for sub-series can be selected to improve the prediction accuracy of the hybrid model. The prediction accuracy of OVMD-BP-QL is better than that of OVMD-LSTM-QL, which is consistent with the conclusion in part 3.5.1. The prediction performance of VMD-FDE-QL is worse than that of OVMD-FDE-QL, which shows that it is not appropriate to select the parameters of VMD only by experience. The randomness of the parameters selected by experience will weaken the final prediction ability of the model. The proposed IAOA-VMD algorithm selects the parameters of VMD well and helps OVMD-FDE-QL obtain the best prediction performance. OVMD-FDE does not select the QL algorithm to ensemble the prediction results of each sub-series, which makes its prediction performance worse than OVMD-FDE-QL. OVMD-FDE-QL effectively ensembles the prediction results of high-frequency and low-frequency components through the QL algorithm, which further improves the prediction results of the model.

3.7 Statistical Test

To further evaluate the effectiveness of the OVMD-FDE-QL, the Diebold–Mariano test [48] is utilized in this section to test whether there are significant differences between the

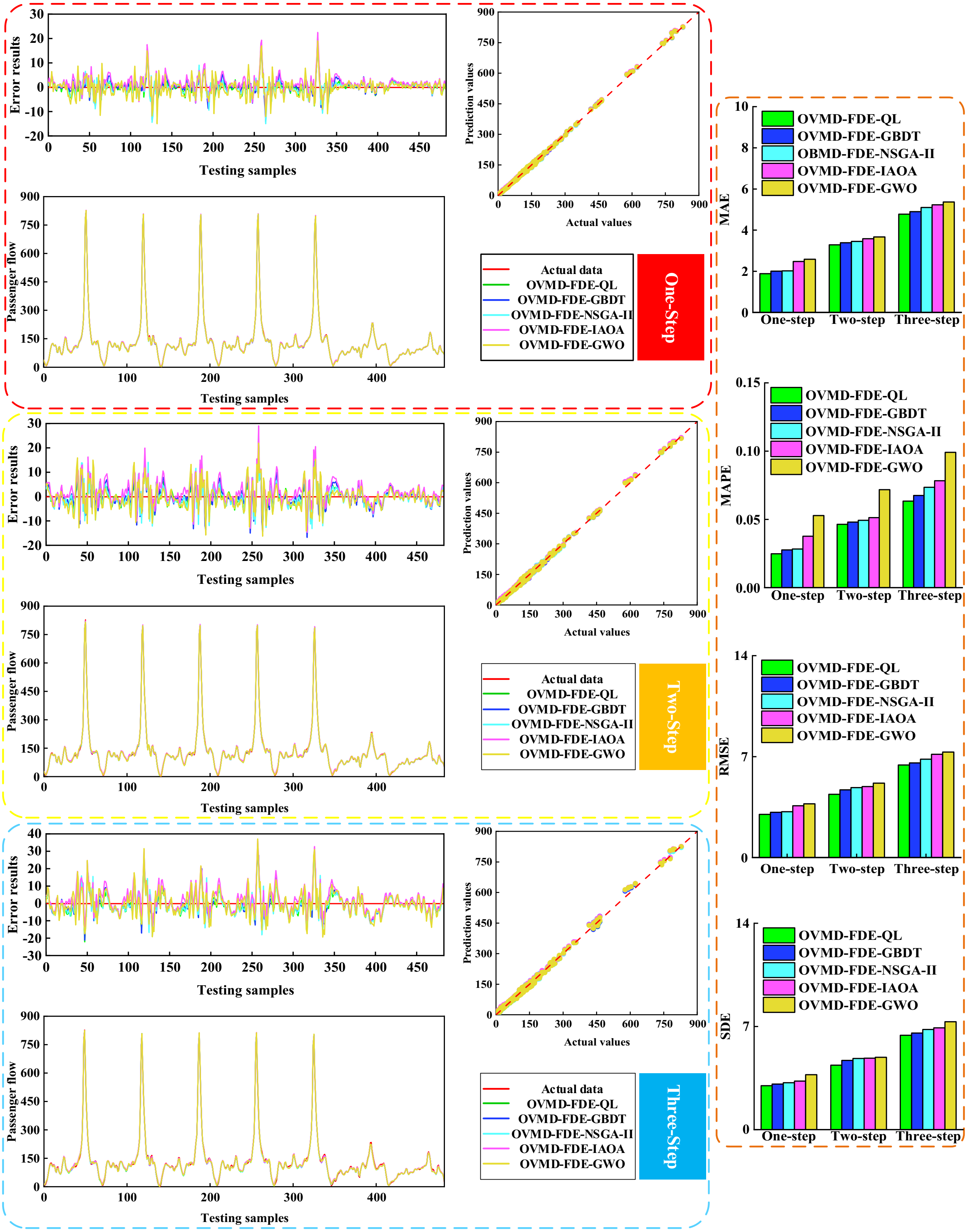


Fig. 11 Results of each models in JHSS in experiment 4

Table 6 Comparison of passenger flow prediction performance of different variant models

Dataset	Model	One-step				Two-step				Three-step			
		MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE	MAE	MAPE	RMSE	SDE
CYSS	**OVMD-FDE-QL**	**2.975**	**0.075**	**3.759**	**3.405**	**3.456**	**0.042**	**4.345**	**4.319**	**6.231**	**0.071**	**7.954**	**7.031**
	OVMD-BP-QL	3.854	0.117	4.726	4.598	5.123	0.062	6.29	6.282	9.116	0.131	11.014	10.959
	OVMD-LSTM-QL	5.32	0.141	9.741	9.192	6.386	0.061	11.051	9.893	9.056	0.143	12.602	12.218
	VMD-FDE-QL	10.275	0.085	12.949	12.937	12.263	0.107	15.159	15.143	15.361	0.152	18.761	18.741
	OVMD-FDE	3.037	0.115	3.825	3.754	4.797	0.055	5.899	5.156	7.094	0.107	8.921	8.906
NNERS	**OVMD-FDE-QL**	**2.695**	**0.198**	**3.57**	**3.564**	**5.439**	**0.242**	**6.931**	**6.93**	**10.44**	**0.411**	**13.039**	**13.037**
	OVMD-BP-QL	5.92	0.359	10.335	10.308	6.364	0.339	10.67	10.644	9.758	0.458	12.839	12.823
	OVMD-LSTM-QL	7.991	0.247	15.333	14.438	8.311	0.279	15.483	14.884	9.368	0.334	16.181	15.672
	VMD-FDE-QL	22.652	0.846	34.666	34.64	22.826	0.782	35.025	35.006	22.428	0.836	34.002	34.001
	OVMD-FDE	2.73	0.21	3.697	3.678	5.655	0.268	7.24	7.228	11.207	0.477	14.351	14.35
NNRS	**OVMD-FDE-QL**	**2.136**	**0.015**	**2.673**	**2.669**	**3.961**	**0.024**	**4.854**	**4.854**	**5.818**	**0.036**	**7.364**	**7.162**
	OVMD-BP-QL	3.148	0.02	3.928	3.248	4.635	0.029	5.997	5.995	6.57	0.043	8.421	8.406
	OVMD-LSTM-QL	2.883	0.029	4.003	3.994	5.516	0.032	6.758	6.046	8.233	0.044	9.989	9.045
	VMD-FDE-QL	12.189	0.075	15.798	15.796	12.761	0.077	16.64	16.636	13.287	0.077	17.114	17.112
	OVMD-FDE	2.913	0.017	3.828	2.816	4.029	0.025	5.107	5.026	6.4	0.039	7.931	7.926
JHSS	**OVMD-FDE-QL**	**1.88**	**0.025**	**2.998**	**2.985**	**3.287**	**0.046**	**4.379**	**4.374**	**4.771**	**0.063**	**6.413**	**6.4**
	OVMD-BP-QL	2.859	0.028	3.957	3.491	3.207	0.05	5.233	4.89	4.693	0.063	7.529	7.227
	OVMD-LSTM-QL	3.106	0.032	4.576	3.918	6.835	0.062	10.042	8.732	9.723	0.119	14.932	13.93
	VMD-FDE-QL	2.394	0.038	3.181	3.184	2.989	0.048	3.883	3.882	3.574	0.051	4.521	4.524
	OVMD-FDE	2.866	0.033	4.604	4.209	3.95	0.059	6.397	6.019	5.582	0.08	8.975	8.495

OVMD-FDE-QL and the comparison models used in the experiment in Sect. 3.4 under a certain confidence level. The DM values predicted in three steps are shown in Table 7, which indicates that except for the ensemble model based on GBDT and NSGA-II, the DM test values of OVMD-FDE-QL and other comparison models are within the rejection range of 1%. Therefore, OVMD-FDE-QL has significant predictive performance at the 1% significance level compared with the standard single models and other hybrid models. Compared with OVMD-FDE-GBDT and OVMD-FDE-NSGA-II, OVMD-FDE-QL can still expressively enhance prediction accuracy at a 5% or 10% significance level. In conclusion, the established OVMD-FDE-QL model has better prediction accuracy, which other models still need to achieve.

3.8 Prediction Performance in Different Time Granularities

The time granularity used in existing studies for STPFP varies in length, and most studies do not consider whether the time granularity used is reasonable. Moreover, for the STPFP at the station level, more emphasis is placed on exemplary management. To ensure prediction accuracy, the appropriate time granularity should be selected as far as possible for STPFP. To compare the prediction performance of the proposed model under different time granularities, the AFC swiping card data were aggregated to 5 minutes and 10 minutes for experiments and compared with the experiment with a 15-minute time granularity. Due to different time granularities, the observed values of the data could be more consistent. Therefore, the correlation coefficient is used as the evaluation index of the model in this section, as shown in Eq. 31:

$$R^2 = 1 - \frac{\sum_{t=1}^{N} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{N} (y_t - \overline{y}_t)^2}, \tag{31}$$

where $\hat{y}_t$ is the predicted value, y_t is the true value, $\overline{y}_t$ is the average value of all true values, and N is the number of samples. Some models in the comparison model are used to compare with OVMD-FDE-QL. The experimental results are shown in Table 8, in which OVMD-FDE-QL has better prediction performance under three different time granularities in four datasets. In addition, with the increase in time granularity, the R^2 value of each model is also increasing, indicating that the prediction accuracy is also increasing with the increase in time granularity. From a statistical point of view, although the passenger flow series extracted at a shorter time granularity can describe the refined information of passenger flow, the regularity is poor, and the prediction accuracy is low. Although the passenger flow series extracted at a more significant time granularity will lose the detailed information on passenger flow. It has

Table 7 Diebold–Mariano test results

Model	Dataset CYSS			Dataset NNERS			Dataset NNRS			Dataset JHSS		
	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step	1-step	2-step	3-step
TCN	9.09*	7.744*	7.431*	7.785*	9.437*	9.952*	11.966*	12.747*	14.099*	4.869*	3.519*	3.51*
BP	11.294*	8.937*	8.545*	9.577*	10.124*	11.546*	14.764*	12.752*	15.263*	7.253*	4.873*	5.401*
LSTM	8.634*	7.666*	7.787*	7.953*	8.999*	10.097*	12.567*	11.744*	12.706*	5.492*	3.873*	4.21*
BP-LSTM-QL	9.642*	7.441*	7.97*	8.06*	9.05*	10.487*	13.004*	11.935*	13.498*	5.576*	4.141*	4.24*
LSTM-TCN-QL	8.986*	7.225*	7.889*	7.396*	8.912*	10.046*	12.448*	11.753*	13.098*	5.336*	3.428*	3.816*
OVMD-BP	8.132*	9.09*	8.978*	5.118*	4.249*	2.093**	10.839*	5.955*	7.435*	7.149*	4.72*	4.522*
OVMD-LSTM	4.842*	5.298*	5.252*	7.261*	7.148*	6.326*	7.452*	12.06*	12.48*	5.065*	7.198*	7.33*
OVMD-FDE	2.363**	8.594*	4.919*	1.687***	2.445**	3.319*	6.248*	2.195**	1.763***	4.966*	4.558*	4.339*
EMD-FDE-QL	8.592*	10.371*	11.019*	10.64*	9.056*	10.583*	6.099*	7.039*	8.702*	4.494*	3.875*	5.803*
EEMD-FDE-QL	11.783*	9.125*	8.563*	6.474*	8.251*	10.571*	8.973*	9.641*	10.228*	4.559*	4.476*	6.228*
CEEMDAN-FDE-QL	6.441*	7.214*	7.685*	10.835*	10.215*	10.723*	7.125*	7.686*	9.311*	5.588*	4.253*	5.37*
WD-FDE-QL	9.297*	10.995*	10.567*	6.235*	9.002*	6.742*	11.221*	12.984*	11.473*	5.373*	7.284*	8.059*
OVMD-FDE-GBDT	2.389**	3.173**	2.762**	2.694**	2.153**	1.561***	6.678*	6.016*	3.028*	1.775***	3.985*	1.794***
OVMD-FDE-NSGA-II	5.263*	6.535*	2.591**	1.556***	1.599***	1.714***	9.862*	7.151*	4.976*	2.222**	3.467*	3.019*
OVMD-FDE-IAOA	9.733*	10.253*	6.291*	4.708*	3.266*	4.159*	10.141*	7.869*	4.984*	6.301*	5.512*	4.456*
OVMD-FDE-GWO	10.45*	9.791*	7.951*	11.945*	7.211*	4.151*	11.822*	8.09*	5.491*	5.346*	3.32*	5.501*

*1% significance level, **5% significance level, and ***10% significance level.

Table 8 Comparison of prediction precision obtained using different time granularities

Dataset	Model	5 min R^2	10 min R^2	15 min R^2
CYSS	**OVMD-FDE-QL**	**0.9586**	**0.9791**	**0.9881**
	LSTM-TCN-QL	0.8441	0.8642	0.8940
	OVMD-FDE	0.9436	0.9619	0.9670
	WD-FDE-QL	0.9251	0.9354	0.9440
	OVMD-FDE-GBDT	0.9351	0.9654	0.9740
NNERS	**OVMD-FDE-QL**	**0.9581**	**0.9643**	**0.9699**
	LSTM-TCN-QL	0.8443	0.8611	0.8831
	OVMD-FDE	0.9440	0.9585	0.9616
	WD-FDE-QL	0.9370	0.9426	0.9444
	OVMD-FDE-GBDT	0.9551	0.9587	0.9683
NNRS	**OVMD-FDE-QL**	**0.9512**	**0.9540**	**0.9556**
	LSTM-TCN-QL	0.8372	0.8641	0.8854
	OVMD-FDE	0.9043	0.9176	0.9338
	WD-FDE-QL	0.8943	0.9116	0.9237
	OVMD-FDE-GBDT	0.9173	0.9226	0.9443
JHSS	**OVMD-FDE-QL**	**0.9667**	**0.9745**	**0.9922**
	LSTM-TCN-QL	0.9546	0.8873	0.9014
	OVMD-FDE	0.9437	0.9548	0.9713
	WD-FDE-QL	0.9316	0.9426	0.9558
	OVMD-FDE-GBDT	0.9601	0.9713	0.9871

strong regularity and high prediction accuracy. Therefore, the selection of time granularity is not smaller, better, or larger. It is necessary to balance the details of passenger flow, regularity, and prediction accuracy. A reasonable time granularity is conducive to capturing more detailed passenger flow information and obtaining acceptable prediction accuracy. According to the experimental results in Table 8, the 15-minute time granularity used in this paper is reasonable given the research background.

4 Conclusions

With the continuous growth of passenger demand, ultra-large-scale passenger flow is becoming more and more normalized. Accurate STPFP can effectively alleviate the operational pressure of the URT. Aiming at the difficulty of STPFP caused by nonlinear and nonstationary passenger flow time series at URT stations, a model called OVMD-FDE-QL based on optimized VMD hybrid ensemble reinforcement learning is proposed. Specifically, the improved IAOA is used for VMD parameter optimization with envelope entropy as the fitness function. The results show that the IAOA has the best convergence speed and optimization ability compared with other optimization algorithms. Then, the original passenger flow sequence is decomposed by OVMD technology, and the complexity of the sub-series is divided by FDE. BP and LSTM are utilized to predict sub-series. The results of OVMD decomposition show that OVMD can decompose complex time series into a series of stable and regular sub-series. In addition, the threshold sensitivity analysis of FDE shows that FDE can not only reflect the complexity of the signal well but also that it is reasonable to divide the high-frequency and low-frequency signals by FDE. Finally, the QL algorithm is used to integrate the prediction results of each sub-series to obtain the final prediction result. The experimental results show that the QL algorithm has better ensemble learning abilities than several classical ensemble methods. The prediction performance and effectiveness of OVMD-FDE-QL are verified by passenger flow time series data and statistical tests at four stations. The main conclusions are as follows:

1. The proposed IAOA optimization algorithm has better optimization abilities than others. In addition, the excellent performance of the IAOA can be used not only for VMD parameter optimization in this paper but also for other optimization scenarios. VMD optimized by the IAOA has better decomposition abilities than other decomposition algorithms, so the time series can be better predicted. In addition, OVMD can be used not only for time series decomposition but also for signal denoising.
2. Through FDE sub-series partitioning, the model can maximize its ability in the appropriate sub-series. The experimental results not only prove the rationality of FDE as a method to judge the complexity of IMF but also help select subsequent combined prediction models.
3. To better integrate the prediction results of the sub-series, a reinforcement learning ensemble strategy is developed in short-term passenger flow prediction. The experimental results show that the reinforcement learning strategy has better ensemble learning ability than other ensemble methods and can effectively improve the overall prediction performance of the proposed model.
4. Experiments are conducted on four station ridership datasets. The experimental results show that OVMD-FDE-QL has good prediction accuracy and significant advantages. In addition, the excellent universality of the model for different subway stations indicates that OVMD-FDE-QL can be used not only for short-term passenger flow prediction but also for other time series prediction fields, such as wind speed prediction and PM2.5 prediction.

However, this study also has limitations. The correlation between URT stations is not considered in the study. Future research will attempt to address these limitations.

Acknowledgements The research was supported by the National Natural Science Foundation of China (Grant No. 52072081), Major Project of Science and Technology of Guangxi Province of China (Grant No. Guike AA20302010), Interdisciplinary Scientific Research Foundation of Guangxi University (Grant No. 2022JCA003), and Innovation Project of Guangxi Graduate Education (Grant No. YCSW2023086).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zhang JP, Wang FY, Wang KF, Lin WH, Xu X, Chen C (2011) Data-driven intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 12(4):1624–1639
2. Ding C, Duan JX, Zhang YR, Wu XK, Yu GZ (2018) Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility. IEEE Trans Intell Transp Syst 19(4):1054–1064
3. Yu L, Feng T, Li T, Cheng L (2023) Demand prediction and optimal allocation of shared bikes around urban rail transit stations. Urban Rail Transit 9(1):57–71
4. Bezuglov A, Comert G (2016) Short-term freeway traffic parameter prediction: application of grey system theory models. Expert Syst Appl 62:284–292
5. Smith BL, Demetsky MJ (1997) Traffic flow forecasting: comparison of modeling approaches. J Transp Eng 123(4):261–266
6. Jiao PP, Li RM, Sun T, Hou ZH, Ibrahim A (2016) Three revised Kalman filtering models for short-term rail transit passenger flow prediction. Math Probl Eng 2016:9717582
7. Ding C, Wang DG, Ma XL, Li HY (2017) Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. Sustainability 8(11):1100
8. Yan H, Fu LY, Qi Y, Yu DJ, Ye QL (2022) Robust ensemble method for short-term traffic flow prediction. Future Gener Comput Syst 133:359–410
9. Liu RJ, Wang YH, Zhou H, Qian ZQ (2019) Short-term passenger flow prediction based on wavelet transform and kernel extreme learning machine. IEEE Access 7:158025–158034
10. Li HY, Wang YT, Xu XY, Qin LQ, Zhang HY (2019) Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network. Appl Soft Comput 83:105620
11. Huang WH, Song GJ, Hong HK, Xie KQ (2014) Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15(5):2191–2201
12. Lv YS, Duan YJ, Kang WW, Li ZX, Wang FY (2014) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
13. Liu DQ, Wu Z, Sun SR (2022) Study on subway passenger flow prediction based on deep recurrent neural network. Multimed Tools Appl 81(14):18979–18992
14. Liu JX, Jiang R, Zhu D, Zhao JD (2022) Short-term subway inbound passenger flow prediction based on AFC Data and PSO-LSTM optimized model. Urban Rail Transit 8(1):56–66
15. Ma XL, Zhang JY, Du BW, Ding C, Sun LL (2018) Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction. IEEE Trans Intell Transp Syst 20(6):2278–2288
16. Wu JX, Li XW, He DQ, Li Q, Xiang WB (2023) Learning spatial-temporal dynamics and interactivity for short-term passenger flow prediction in urban rail transit. Appl Intell 53(16):19785–19806
17. Cai LR, Lei MQ, Zhang SY, Yu YD, Zhou T, Qin J (2020) A noise-immune LSTM network for short-term traffic flow forecasting. Chaos 30(2):023135
18. Chen XQ, Lu JQ, Zhao JS, Qu ZJ, Yan YS, Xian JF (2020) Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network. Sustainability 12(9):3678
19. Wei Y, Chen MC (2012) Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transp Res Part C Emerg Technol 21(1):148–162
20. Yang HF, Chen YPP (2019) Hybrid deep learning and empirical mode decomposition model for time series applications. Expert Syst Appl 120:128–138
21. Li YP, Ma CX (2023) Short-time bus route passenger flow prediction based on a secondary decomposition integration method. J Transp Eng Part A Syst 149:04022132
22. Zhang SC, Zhou LX, Chen XQ, Zhang L, Li L, Li M (2020) Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition. Comput Aided Civ Infrastruct Eng 35(10):1132–1147
23. Liu J, Wu NQ, Qiao Y, Li ZW (2022) Short-term traffic flow forecasting using ensemble approach based on deep belief networks. IEEE Trans Intell Transp Syst 23(1):404–417
24. Jia XL, Zhou WX, Yang HZ, Li SQ, Chen XP (2023) Short-term traffic travel time forecasting using ensemble approach based on long short-term memory networks. IET Intell Transp Syst 17(6):1262–1273
25. Huang H, Mao JN, Lu WK, Hu GJ, Liu L (2023) DEASeq2Seq: An attention based sequence to sequence model for short-term metro passenger flow prediction within decomposition-ensemble strategy. Transp Res Part C Emerg Technol 146:103965
26. Diao ZL, Zhang DF, Wang X, Xie K, He SY, Lu X, Li YB (2018) A hybrid model for short-term traffic volume prediction in massive transportation systems. IEEE Trans Intell Transp Syst 20(3):935–946
27. Liu Y, Song YL, Zhang Y, Liao ZF (2022) WT-2DCNN: A convolutional neural network traffic flow prediction model based on wavelet reconstruction. Physica A 603:127817
28. Yang X, Xue QC, Yang XX, Yin HD, Qu YC, Li X, Wu JJ (2021) A novel prediction model for the inbound passenger flow of urban rail transit. Inf Sci 566:347–363
29. Dragomiretskiy K, Zosso D (2014) Variational mode decomposition. IEEE Trans Signal Process 62(3):531–544

30. Li HT, Jin K, Sun SL, Jia XY, Li YW (2022) Metro passenger flow forecasting though multi-source time-series fusion: An ensemble deep learning approach. Appl Soft Comput 120:108644
31. Li J, Zhang ZC, Meng FX, Zhu W (2022) Short-term traffic flow prediction via improved mode decomposition and self-attention mechanism based deep learning approach. IEEE Sens J 22(14):14356–14365
32. Yang H, Cheng YX, Li GH (2022) A new traffic flow prediction model based on cosine similarity variational mode decomposition, extreme learning machine and iterative error compensation strategy. Eng Appl Artif Intell 115:105234
33. Zhang YY, Zhu CF, Wang QR (2020) LightGBM-based model for metro passenger volume forecasting. IET Intell Transp Syst 14(13):1815–1823
34. Fu WL, Wang K, Li CS, Tan JW (2019) Multi-step short-term wind speed forecasting approach based on multi-scale dominant ingredient chaotic analysis, improved hybrid GWO-SCA optimization and ELM. Energy Convers Manag 187:356–377
35. Huang YS, Gao YL, Gan Y, Ye M (2021) A new financial data forecasting model using genetic algorithm and long short-term memory network. Neurocomputing 425:207–218
36. Liu Q, Liu M, Zhou HL, Yan F (2022) A multi-model fusion based non-ferrous metal priceforecasting. Resour Policy 77:102714
37. Li GH, Zheng CF, Yang H (2022) Carbon price combination prediction model based on improved variational mode decomposition. Energy Rep 8:1644–1664
38. Yang K, Wang BF, Qiu X, Li JH, Wang YZ, Liu YL (2022) Multi-step short-term wind speed prediction models based on adaptive robust decomposition coupled with deep gated recurrent unit. Energies 15(12):4221
39. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
40. Goh HH, He RH, Zhang DD, Liu H, Dai W, Lim CS, Kurniawan TA, Teo KTK, Goh KC (2022) A multimodal approach to chaotic renewable energy prediction using meteorological and historical information. Appl Soft Comput 118:108487
41. Sun W, Duan M (2019) Analysis and forecasting of the carbon price in China's regional carbon markets based on fast ensemble empirical mode decomposition, phase space reconstruction, and an improved extreme learning machine. Energies 12(2):277
42. Li HT, Jin F, Sun SL, Li YW (2021) A new secondary decomposition ensemble learning approach for carbon price forecasting. Knowl Based Syst 214:106686
43. Azami H, Escudero J (2018) Amplitude- and fluctuation-based dispersion entropy. Entropy 20(3):210
44. Chen C, Liu H (2021) Dynamic ensemble wind speed prediction model based on hybrid deep reinforcement learning. Adv Eng Inform 48:101290
45. Dong YC, Zhang HL, Wang C, Zhou XJ (2021) Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. Neurocomputing 462:169–184
46. Liu H, Deng DH (2022) An enhanced hybrid ensemble deep learning approach for forecasting daily PM2.5. J Cent South Univ 29(6):2074–2083
47. Tian ZD (2021) Modes decomposition forecasting approach for ultra-short-term wind speed. Appl Soft Comput 105:107303
48. Song JJ, Wang JZ, Lu HY (2018) A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. Appl Energy 215:643–658
49. Cheng ZS, Wang JY (2020) A new combined model based on multi-objective salp swarm optimization for wind speed forecasting. Appl Soft Comput 92:106294
50. Chen J, Liu H, Chen C, Duan Z (2022) Wind speed forecasting using multi-scale feature adaptive extraction ensemble model with error regression correction. Expert Syst Appl 207:117358
51. Liu H, Yu CQ, Wu HP, Duan Z, Yan GX (2020) A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. Energy 202:117794
52. Jin ZZ, He DQ, Lao ZP, Wei ZX, Yin XH, Yang WF (2023) Early intelligent fault diagnosis of rotating machinery based on IWOA-VMD and DMKELM. Nonlinear Dyn 111(6):5287–5306
53. Liu Y, Qin Z, Liao XF, Wu JH (2020) Cryptanalysis and enhancement of an image encryption scheme based on a 1-D coupled Sine map. Nonlinear Dyn 100(3):2917–2931
54. He DQ, Liu CY, Jin ZZ, Ma R, Chen YJ, Shan S (2022) Fault diagnosis of flywheel bearing based on parameter optimization variational mode decomposition energy entropy and deep learning. Energy 239:122108
55. Liang T, Lu H, Sun HX (2021) Application of parameter optimized variational mode decomposition method in fault feature extraction of rolling bearing. Entropy 23(5):520
56. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
57. Sun W, Tan B, Wang QQ (2021) Multi-step wind speed forecasting based on secondary decomposition algorithm and optimized back propagation neural network. Appl Soft Comput 113:107894
58. Wiering MA, Otterlo MV (2012) Reinforcement learning. Adapt Learn Optim 12(3):729
59. Szepesvári C (2010) Algorithms for reinforcement learning. Synth Lect Artif Intell Mach Learn 4(1):1–103
60. Li XW, Huang ZX, Liu SH, Wu JX, Zhang YX (2023) Short-term subway passenger flow prediction based on time series adaptive decomposition and multi-model combination (IVMD-SE-MSSA). Sustainability 15(10):7949
61. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Proceedings of the international workshop on nature inspired cooperative strategies for optimization, pp 65–74
62. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197